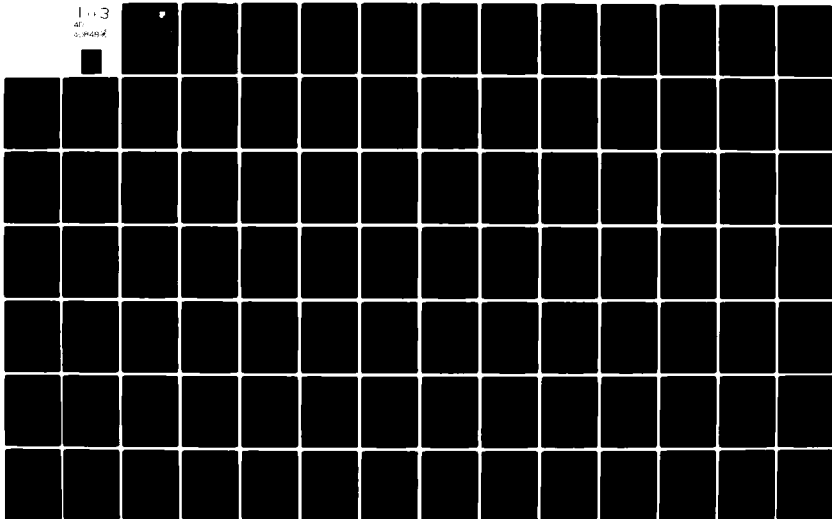


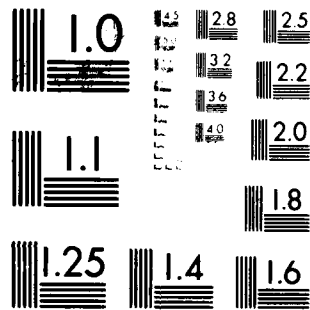
AD-A084 895

MEASUREMENT CONCEPT CORP ROME NY F/G 17/9
TACTICAL TARGET IDENTIFICATION (TTI) LABORATORY SIMULATION. VOL--ETC(U)
MAR 80 R KELLOGG, R A MCKIE, J TINS F30602-79-C-0029
RADC -TR-80-41-VOL-1 NL

UNCLASSIFIED

1-3
AD
0-140-2





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

ADA 084895

LEVEL

TTI LABORATORY SIMULATION

Measurement Concept Corporation

Richard Kellogg
Robert A. McKie
James Tims

DTIC
ELECTRA
MAY 27 1980

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

THIS DOCUMENT IS BEST QUALITY PRACTICES.
THE COPY FURNISHED TO DDC CONTAINED A
SIGNIFICANT NUMBER OF PAGES WHICH DO NOT
REPRODUCE LEGIBLY.

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

FILE COPY

80 5 27 231

This report was reviewed by the RADC Plans Office and is releasable to the National Technical Information Service (NTIS). Volume II contains only the information which is releasable to the general public.

This report has been reviewed by the RADC Plans Office and is releasable to the National Technical Information Service (NTIS). It will be releasable to the general public, including foreign nations.

RADC-TR-80-41, Volume I (of two) has been reviewed and is approved for publication.

APPROVED:



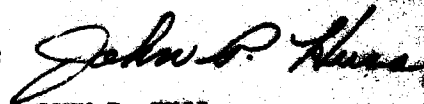
RICHARD J. WOOD
Project Engineer

APPROVED:



FRANK J. REHM
Technical Director
Surveillance Division

FOR THE COMMANDER:



JOHN P. HUSS
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (OCTM) Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return this copy. Retain or destroy.

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

19 REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM	
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER	
(18) RADC-TR-80-41-Vol-1 (of two)	AD-A081895		
4. TITLE AND SUBTITLE		5. TYPE OF REPORT & PERIOD COVERED	
(6) Tactical Target Identification (TTI) LABORATORY SIMULATION, Volume I.		Final Technical Report, Nov 78 - Oct 79	
6. AUTHOR		6. PERFORMING ORG. REPORT NUMBER	
(10) Richard/Kellogg Robert A./McKie James/Tims		N/A	
7. PERFORMING ORGANIZATION NAME AND ADDRESS		8. CONTRACT OR GRANT NUMBER(s)	
Measurement Concept Corporation Rome NY 13440		(15) F30602-79-C-0029 / new	
9. CONTROLLING OFFICE NAME AND ADDRESS		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS	
Rome Air Development Center (OCTM) Griffiss AFB NY 13441		62702F (16) 45061341 (12) 13	
11. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE	
Same (13) 278		Mar 1980	
13. SECURITY CLASS. (of this report)		13. NUMBER OF PAGES	
UNCLASSIFIED		269	
14. DISTRIBUTION STATEMENT (of this Report)		15. SECURITY CLASS. (of this report)	
Approved for public release; distribution unlimited.		UNCLASSIFIED	
16. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	
Same		N/A	
18. SUPPLEMENTARY NOTES			
RADC Project Engineer: Richard J. Wood (OCTM)			
Vol II of this report will not be published for routine (see reverse)			
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)			
Simulation Image Conversions			
Processing Interactive			
Identification Programs			
Radar Display			
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)			
Under the subject contract a general capability was developed to allow TTI Laboratory users to communicate with the COMTAL Image Processor, both under the DEC operating systems RT-11 and RSW-11M. This was specifically accomplished in three phases. First, the COMTAL Vision One Image Processing System was made accessible to the PDP-11/40 Processor under the RT-11 single-user operating system. Second, the COMTAL Image Processor was made to operate under the RSW-11M multi-user			

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

393 365

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

operating system, and was then integrated with the existing radar image processing system. Third, the radar return and air frame display program, Silhouette, was successfully converted to RSX-11M, and software written to enable the COMTAL image display to replace the previously supported DEC VT-11 Calligraphic Display.

Block 18 Cont'd

distribution but is available from RADC (OCTM), Griffiss AFB NY 13441 upon request. Volume II contains only the program listings as pertains to this volume.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

PREFACE

This document describes technical services provided to Rome Air Development Center by the Measurement Concept Corporation under Contract F30602-79-C-0029. Three distinct efforts were performed by Mc² to integrate the COMTAL Vision One Image Display System with existing tools present in RADC's TTI (Tactical Target Identification) laboratory, located in Building 112 at GAFB. The Project Engineer of the subject effort was Mr. Richard Wood.

Accession For	
MAIL ROOM	<input checked="checked" type="checkbox"/>
DDC R&B	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	<input type="checkbox"/>
By	
Distribution/	
Availability Codes	
Dist.	Avail and/or special
A23	P

TABLE OF CONTENTS

	<u>Page</u>
1.0 RT-11COMTAL TO PDP-11 COMMUNICATIONS UTILITY.	1-1
1.1 COMPIP.FOR	1-2
1.2 CPIPCB.FOR	1-2
1.3 CPIPLB.FOR	1-2
1.4 COMLØ1.FOR	1-3
1.5 COMLØ2.FOR	1-3
1.6 COMLØ3.FOR	1-4
1.7 COMLØ4.FOR	1-4
1.8 COMLØ5.FOR	1-5
1.9 Instructions for Building the COMPIP Task	1-5
1.10 To Run COMPIP	1-6
2.0 INTEGRATION OF THE COMTAL VISION ONE WITH THE RIPS (RADAR) IMAGE PROCESSING SYSTEM	2-1
2.1 FRAMES.SCR and FRAMES.IPS	2-3
2.2 CALSUB.FTN	2-4
2.3 EXEC.ODL	2-4
2.4 PIPSB.FTN	2-4
2.5 COMLØ1.FOR	2-5
2.6 COMLØ2.FOR	2-5
2.7 COMLØ3.FOR	2-6
2.8 COMLØ4.FOR	2-6
2.9 COMLØ5.FOR	2-7
2.10 IWRTW.MAC	2-8
2.11 Building the Modified RIPS Task	2-9
2.12 To Run "RIPS" With The COMTAL	2-9
2.13 Integration of Mc ² Software With Other User's Program	2-9

TABLE OF CONTENTS (Continued)

	<u>Page</u>
3.0	COMTAL: MC ² VERSION OF THE SRC SILHOUETTE PROGRAM . . 3-1
3.1	Operation of the New SILHOUETTE Program 3-2
3.1.1	Preparation for Execution 3-2
3.1.2	Preparation for Execution of SILHOUETTE Program . . . 3-4
3.1.3	Operator Interaction with the SILHOUETTE Program. . . 3-5
3.1.4	Building "COMTAL", The Mc ² Version of "SILHOUETTE". . 3-7
3.2	Modification and Additions to SRC Software 3-7
3.2.1	Modifications to SLMAIN.FTN 3-7
3.2.2	Modification to SLSUBS.FTN 3-10
3.2.3	COMGRF.FTN 3-11
3.2.4	Subroutine ROTATE 3-13
3.2.5	VECTOR.FTN 3-14
3.2.6	Other Programs 3-14

SUMMARY

The technical services provided by Mc² to RADC under the subject contract added three tools to the Tactical Target Identification Laboratory (TTIL) in Building 112. First, the COMTAL Vision One Image Processing System was made accessible to the PDP-11/40 processor under the RT-11 Single-User operating system. Second, the COMTAL Vision One was made to operate under the RSX-11M Multi-User Operating System, and was then integrated with the existing PAR-developed Radar Image Processing System. Third, the Syracuse Research Corporation's Radar Return and Airframe Display Program, SILHOUETTE, was successfully converted to RSX-11M, and software was written to allow the COMTAL device to replace the previously supported DEC VT-11 Calligraphic Display.

In addition to these specific demonstrable tools, Mc² is leaving behind a general capability that will allow future TTIL users to easily communicate with the COMTAL system, both under RT-11 and under RSX-11M.

The following documentation describes the three main efforts under this contract. Copies of all software discussed reside both on RK-05 disk packs, and on backup tape volumes.

Many of the difficult problems encountered under this contract required trial-and-error solution, and the future avoidance of these problems by future TTIL users is an important legacy of this contract.

Because of the disjoint nature of the three tasks funded by this contract, this Final Technical Report will be divided into three sections, each describing a complete task. Interrelationships between these tasks, where they exist, are duly noted, and there is a certain amount of back-and-forward referencing between sections. The three subtasks were carried

out chronologically in the order presented. Each task was executed by a different member of Mc²'s technical staff. The assignments are as follows:

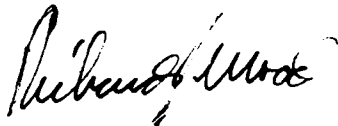
COMPIP:	Richard Kellogg
RIPS, RSX and COMTAL Conversion:	James Tims
SILHOUETTE Conversion:	Robert A. McKie

In conclusion, Mc² would like to express its appreciation of the assistance given by the staff of the TTI Laboratory, Mr. P. Grasso, Mr. N. Internicola, and particularly Mr. Ronald Blackall, who was a concerned source of unwritten documentation, and a co-equal contributor to the three efforts.

Robert A. McKie, Project Manager
Measurement Concept Corporation

EVALUATION

The Tactical Target Identification (TTI) Laboratory/Simulation program was instituted to exploit the hardware/software resources of the TTI Laboratory. The evolution of effective peripherals for Target Identification Simulation and Real Data Processing, required an equivalent upgrade of software capabilities, which included modifications and development of an effective operating system controller. This effort provided for the combining of various independent programs for more effective operator control, and established existing capabilities from one hardware system to a more effective peripheral. The capabilities of the TTI Laboratory have thus been enhanced in performance, adaptability to new programs, and ease of understanding for future TTI applications. This effort was in accordance with TPO R2E to establish simulation capabilities for evaluation and validation of system programs for TTI assessments.



RICHARD J. WOOD
Project Engineer

1.0 RT-11 COMTAL TO PDP-11 COMMUNICATIONS UTILITY

The initial major task of the subject contract involved constructing a software package that would utilize the COMTAL DOS (Disk Operating System) device driver. This driver was modified by R. Kellogg to be usable under RT-11, but was tested initially on single lines of data. The objective of the initial effort was to construct a utility that would permit transfer of existing RIPS (Radar Image Processing System) files to the COMTAL for display.

The resulting COMPIP (COMTAL PIP) system served a twofold purpose. First, it allowed verification and demonstration of the VISION ONE by providing it a source of "real" data. Second, COMPIP was designed in modules that would be relatively easy to convert to RSX-11M. This design supported the objective of adapting the RIPS system to use the COMTAL for graphic output which, in fact, was the second major task undertaken, described in Section 2.0.

As constructed, COMPIP is a stand-alone program that functions much the same as the DEC PIP (Peripheral Interchange Program). It displays a list of COMTAL display options, requests a user selection from this list, then prompts the user into entering both file identification information and COMTAL memory identification information. Once entered, the information so described is written to or from the COMTAL.

Although not designed to be used as a tool integratable with a user application, COMPIP can indeed be used as part of a user program by removing the COMPIP.FOR main program, and building the remaining modules into the user's task.

The following paragraphs describe the functions of each of the software modules contained in COMPIP.

1.1 COMPIP.FOR

Author: R. Kellogg, Measurement Concept Corp., 1979.

COMPIP is the main program of the COMPIP task. It uses FORTRAN Read/Write statements to query the user regarding which COMTAL function he would like performed, then directs passage of control to the proper subroutine contained in CPIPSB.FOR. This program is essentially a "dummy" main calling program.

1.2 CPIPSB.FOR

CPIPSB.FOR contains the main subroutines for the program COMPIP. Each subroutine is a self-contained main section, and calls several of the COMTAL library routines, as well as other IPS file reading routines. The main routines in CPIPSB.FOR are:

- IWIMF - Read an IPS image file to the COMTAL.
- IRIMF - Write an IPS image file from the COMTAL.
- IWGMF - Read an IPS graphics file to the COMTAL.
- IRGMF - Write an IPS graphics file from the COMTAL.
- IWFMF - Read an IPS function file to the COMTAL.
- IRFMF - Write an IPS function file from the COMTAL.
- IWPSF - Read an IPS pseudo-color file to the COMTAL.
- IRPSF - Write an IPS pseudo-color file from the COMTAL.

1.3 CPIPLB.FOR

CPIPLB.FOR is a collection of subroutines that open, close, read and write various IPS files for use with COMPIP. When COMPIP is later re-written to be integrated as a RIPS "frame" option, the module will not be necessary.

The routines contained in CPIPLB.FOR are:

- I PROPE - Opens an IPS file for reading.
- I PW OPE - Opens an IPS file for output.
- I PR DIM - Reads one record from an IPS file to the user's buffer.
- I PW TIM - Writes one record from the user's buffer to an IPS file.
- I PC LOS - Closes an IPS file.
- I PR DPS - Reads one Pseudo-color Memory record into user buffer.
- I PW TPS - Writes a Pseudo-color Memory record to an IPS file.
- I PR DGM - Reads one line of Graphics Memory from an IPS file into the user buffer.
- I PW TGM - Writes one Graphic Memory record to an IPS file.
- I PR DFM - Reads one Function Memory record from an IPS file to the user buffer.
- I PW TFM - Writes one Function Memory record to an IPS file.

1.4 COMLØ1.FOR

COMLØ1.FOR is the first of five COMTAL library modules containing routines that support reading and writing to and from the COMTAL. COMLØ1 establishes some data definitions, common areas, data equivalences and COMTAL command word tables. It also contains two routines to open and close the COMTAL's I/O channel:

- I CMOPE - Opens channel for COMTAL service.
- I CM CLO - Closes the COMTAL channel.

1.5 COMLØ2.FOR

COMLØ2.FOR contains routines to read and write COMTAL "Image" and "Graphic" memories.

- I WIM - Writes one line of data from user memory to selected COMTAL Image Memory.

- IRIM - Reads one line of data from selected COMTAL Image Memory to user buffer.
- IWGM - Writes one line of data from user memory to the selected COMTAL Graphic Memory.
- IRGM - Reads one line of data from the selected COMTAL Graphic Memory to the user buffer.
- ITC1 - Performs the actual "Transfer Code 1" command for the above four routines.

1.6 COML03.FOR

COML03.FOR contains routines to read and write COMTAL Function and Pseudo-color Memory. It contains the following routines:

- IWFM - Writes to the selected COMTAL Function Memory from the user buffer.
- IRFM - Reads the selected Function Memory into the user buffer.
- IWPS - Writes to the selected Pseudo-color Memory from the user buffer.
- IRPS - Reads from the selected Pseudo-color Memory into the user buffer.
- ITC2 - Performs the actual "Transfer Code 2" command for the above four routines.

1.7 COML04.FOR

COML04.FOR contains routines to read and write the location of the COMTAL cursor. There are only two subroutines in this module. They are:

- WTP - Writes X, or Y, or both to the target cursor in the COMTAL.
- RTP - Reads the target cursor X, Y location into variables IX, IY.

1.8 COML05.FOR

COML05.FOR contains routines to read and write blocks of text to and from the COMTAL, read and write the data block to and from the COMTAL, and read and write blocks of contiguous memory between the COMTAL and the host computer. The routines are:

- WCB - Transmit a block of text to the COMTAL text buffer.
- RCB - Read a block of text from the COMTAL text buffer.
- WDB - Transmit a Data Block to the COMTAL. The Data Block contains the contents of several display registers that determine the status of the COMTAL display.
- RDB - Reads the COMTAL Data Block.
- WCD - This routine is used to write any block of memory in the COMTAL. Its primary use is down line loading code to the COMTAL, when COMTAL memory is acquired to support this loading.
- RCD - This routine reads any block of memory from the COMTAL.
- TC3 - This routine issues the "Transfer Code 3" command for the above six routines.

1.9 Instructions for Building the COMPIP Task

The following RT-11 commands are necessary to instruct the RT-11 Linker to link together the various modules comprising COMPIP into an executable task image:

```
.R LINK
*COMPIP = COMPIP, CPIPSB, CPIPLB, SYSLIB/F/C
*COML01, COML02, COML03 (, COML04, COML05)
```

1.10 To Run COMPIP

.RUN COMPIP

COMPIP will prompt the user for necessary input.


```

002S      GO TO 50
0029      END

```

```

RT-11 FORTRAN IV      STORAGE MAP
NAME      OFFSET      ATTRIBUTES
IRU      001204      INTEGER12 VARIABLE
IUIMF      000000      INTEGER12 PROCEDURE
IRIMF      000000      INTEGER12 PROCEDURE
IUGMF      000000      INTEGER12 PROCEDURE
IUMF      000000      INTEGER12 PROCEDURE
IUFMF      000000      INTEGER12 PROCEDURE
IRPMF      000000      INTEGER12 PROCEDURE
IUPSF      000000      INTEGER12 PROCEDURE
IRPSF      000000      INTEGER12 PROCEDURE
IUCBF      000000      INTEGER12 PROCEDURE
1

```

RT-11 FORTRAN IU U018-08A UED 18-APR-79 18:12:34 PAGE 001

FILE NAME: CIPSB.FOR

DATE LAST EDITED: 18-APR-79

HISTORY:

DATE:	MODIFICATION
19-MAR-79	INITIAL WRITING
21-MAR-79	ADD MORE SUBS
18-APR-79	ADD CLEAR SCREEN
	TO 'IUIWF' AND
	ADD 'IUCBF'

THIS FILE CONTAINS THE MAIN SUBROUTINES FOR THE
PROGRAM 'COMPIP'. EACH SUBROUTINE IS A SELF CONTAINED
MAIN SECTION AND CALLS SEVERAL OF THE CORTAL LIBRARY
ROUTINES AS WELL AS OTHER IPS FILE READING ROUTINES.

THE MAIN ROUTINES ARE:

IUIWF	- READ AN IPS FILE DISPLAY ON CORTAL
IRIWF	- WRITE AN IPS FILE FROM THE CORTAL
IUGMF	- READ AN IPS GRAPHICS FILE TO CORTAL
IRGMF	- WRITE AN IPS GRAPHICS FILE FROM CORTAL
IUFMF	- READ AN IPS FUNCTION FILE TO CORTAL
IRFMF	- WRITE AN IPS FUNCTION FILE FROM CORTAL
IUPSF	- READ AN IPS PSEUDO-COLOR FILE TO CORTAL
IRPSF	- WRITE AN IPS PSEUDO-COLOR FILE FROM CORTAL
IUCBF	- WRITE COMMAND BLOCK TO CORTAL

0001 SUBROUTINE IUIWF

FUNCTIONAL DESCRIPTION:

THIS ROUTINE WILL READ AN IPS 'IMAGE' TYPE FILE AND
WRITE IT TO ONE OF THE IMAGE MEMORIES IN THE CORTAL.

1) QUERY USER FOR IMAGE NUMBER
2) OPEN FILE FOR READ/WRITE
3) OPEN CORTAL FOR PROCESSING

0212
0213

RT-11	FORTAN IV	STORAGE MAP
NAME	OFFSET	ATTRIBUTES
ITEXT	000014	LOGICALX1
INTYP	000024	INTEGERX1
ONE	000030	LOGICALX1
TWO	000031	LOGICALX1
THREE	000032	LOGICALX1
FOUR	000033	LOGICALX1
FIVE	000036	INTEGERX1
SIX	000037	INTEGERX1
SEVEN	000038	INTEGERX1
EIGHT	000039	INTEGERX1
NINE	000040	INTEGERX1
TEN	000041	INTEGERX1
ELEVEN	000042	INTEGERX1
TWELVE	000043	INTEGERX1
THIRTEEN	000044	INTEGERX1
FOURTEEN	000045	INTEGERX1
FIFTEEN	000046	INTEGERX1
SIXTEEN	000047	INTEGERX1
SEVENTEEN	000048	INTEGERX1
EIGHTEEN	000049	INTEGERX1
NINETEEN	000050	INTEGERX1
TWENTY	000051	INTEGERX1
COMMON	000052	COMMON
BLOCK	000053	COMMON
LINE	000054	COMMON
LENGTH	000055	COMMON
ARRAY	000056	COMMON
SIZE	000057	COMMON
INDEX	000058	COMMON
VALUE	000059	COMMON
ADDRESS	000060	COMMON
POINT	000061	COMMON
TO	000062	COMMON
FROM	000063	COMMON
BY	000064	COMMON
NAME	000065	COMMON
VALUE	000066	COMMON
ADDRESS	000067	COMMON
POINT	000068	COMMON
TO	000069	COMMON
FROM	000070	COMMON
BY	000071	COMMON
NAME	000072	COMMON
VALUE	000073	COMMON
ADDRESS	000074	COMMON
POINT	000075	COMMON
TO	000076	COMMON
FROM	000077	COMMON
BY	000078	COMMON
NAME	000079	COMMON
VALUE	000080	COMMON
ADDRESS	000081	COMMON
POINT	000082	COMMON
TO	000083	COMMON
FROM	000084	COMMON
BY	000085	COMMON
NAME	000086	COMMON
VALUE	000087	COMMON
ADDRESS	000088	COMMON
POINT	000089	COMMON
TO	000090	COMMON
FROM	000091	COMMON
BY	000092	COMMON
NAME	000093	COMMON
VALUE	000094	COMMON
ADDRESS	000095	COMMON
POINT	000096	COMMON
TO	000097	COMMON
FROM	000098	COMMON
BY	000099	COMMON
NAME	000100	COMMON
VALUE	000101	COMMON
ADDRESS	000102	COMMON
POINT	000103	COMMON
TO	000104	COMMON
FROM	000105	COMMON
BY	000106	COMMON
NAME	000107	COMMON
VALUE	000108	COMMON
ADDRESS	000109	COMMON
POINT	000110	COMMON
TO	000111	COMMON
FROM	000112	COMMON
BY	000113	COMMON
NAME	000114	COMMON
VALUE	000115	COMMON
ADDRESS	000116	COMMON
POINT	000117	COMMON
TO	000118	COMMON
FROM	000119	COMMON
BY	000120	COMMON
NAME	000121	COMMON
VALUE	000122	COMMON
ADDRESS	000123	COMMON
POINT	000124	COMMON
TO	000125	COMMON
FROM	000126	COMMON
BY	000127	COMMON
NAME	000128	COMMON
VALUE	000129	COMMON
ADDRESS	000130	COMMON
POINT	000131	COMMON
TO	000132	COMMON
FROM	000133	COMMON
BY	000134	COMMON
NAME	000135	COMMON
VALUE	000136	COMMON
ADDRESS	000137	COMMON
POINT	000138	COMMON
TO	000139	COMMON
FROM	000140	COMMON
BY	000141	COMMON
NAME	000142	COMMON
VALUE	000143	COMMON
ADDRESS	000144	COMMON
POINT	000145	COMMON
TO	000146	COMMON
FROM	000147	COMMON
BY	000148	COMMON
NAME	0	

PT-11 FORTRAN IV U018-08A WED 18-APR-79 18:12:38

```

00001
00000000
SUBROUTINE IRINF
,      FUNCTIONAL DESCRIPTION:
,      THIS ROUTINE WILL READ A COMTAL IMAGE AND
,      WRITE IT TO AN IPS ' IMAGE ' FILE.

```

```

0002      1) QUERY USER FOR IMAGE NUMBER
0003      2) OPEN FILE FOR READ/WRITE
0004      3) OPEN COMTAL FOR PROCESSING
0005      4) READ COMTAL, WRITE FILE

      INTEGER NFTYP(2),NAFTYP(2)
      INTEGER IBUFF(512)
      COMMON /LINE/ IBUFF

      DATA NFTYP /3RING,3RING/, NAFTYP /'.I','MG'/

      1) QUERY USER

      WRITE(7,2000)
      FORMAT(' ', INPUT IMAGE NUMBER: ',S)
      READ(7,110)IMNO
      FORMAT(2I6)

      READ COMTAL,WRITE IPS FILE

      1020 CONTINUE
      PAUSE / MAIN 3 JUST PRIOR IPUOPE /

      IER = IPUOPE(256,256,NFTYP,NAFTYP)
      IER-ICHOPE()

      PAUSE / MAIN 4 JUST PRIOR DO LOOP WRITE/READ /
      DO 200 J=1,256
      IER-IRIM(IMNO,J,IBUFF)
      IER = IPUTIM(IBUFF)
      CONTINUE
      CALL IPCLOS
      CALL ICMCLO

```

```

C
0013 RETURN
0020 END

```

```

RT-11 FORTRAN IV      STORAGE MAP
NAME  OFFSET  ATTRIBUTES
NFTYP 000014 INTEGER12 ARRAY (2)
NAFTYP 000020 INTEGER12 ARRAY (2)
IMNO  000070 INTEGER12 VARIABLE
IER   000072 INTEGER12 VARIABLE
IPROPE 000000 INTEGER12 PROCEDURE
ICHOPE 000000 INTEGER12 PROCEDURE
J      000074 INTEGER12 VARIABLE
IRIM   000000 INTEGER12 PROCEDURE
IPUTIM 000000 INTEGER12 PROCEDURE
IPCL0S 000000 INTEGER12 PROCEDURE
ICML0  000000 INTEGER12 PROCEDURE
COMMON BLOCK /LINE/  LENGTH 002000
IRUFF  000000 INTEGER12 ARRAY (512)

```

PT-11 FORTRAN IV U018-08A UED 18-APR-79 18:12:41 PAGE 001

```

0001 SUBROUTINE IUPSF
C
C      FUNCTIONAL DESCRIPTION:
C
C      THIS ROUTINE WILL READ AN IPS PSEUDO-COLOR FILE
C      AND WRITE IT TO THE COMTAL PSEUDO-COLOR MEMORY.
C
C      1) OPEN FILE FOR READ/WRITE
C      2) OPEN COMTAL FOR PROCESSING
C      3) READ FILE, WRITE COMTAL
C      4) READ FILE, WRITE COMTAL
C      5) CLOSE COMTAL, CLOSE IPS FILE
C
0002 INTEGER NFTYP(2)
0003 INTEGER IBUFF(512)

```

```

00014 COMMON /LINE/ IBUFF
00015 DATA NFTYP /3RPSE,3RPSE/
      1)
      2)
00016 IER = IPROPE(NRECS,NRECSZ,NFTYP)
      3) OPEN CONTAL
      IER=ICMOPE( )
      4) READ IPS FILE, WRITE TO CONTAL PSEUDO-COLOR
00017 IER = IPRDPS (IBUFF)
00018 IER = IUPS (IBUFF)
      5) CLOSE IPS FILE, CLOSE CONTAL
      CALL IPCLOS
00019 IER = ICMCLO
      RETURN
00020 END

```

RT-11 FORTRAN IV STORAGE MAP

NAME	OFFSET	ATTRIBUTES
NFTYP	000014	INTEGER*2 ARRAY (2)
IPROPE	000020	INTEGER*2 VARIABLE
IPRDP	000020	INTEGER*2 PROCEDURE
IPRDS	000022	INTEGER*2 VARIABLE
IPRCSZ	000024	INTEGER*2 VARIABLE
ICMOPE	000026	INTEGER*2 PROCEDURE
IPRDP	000028	INTEGER*2 PROCEDURE
IUPS	000030	INTEGER*2 PROCEDURE
IPCLOS	000032	INTEGER*2 PROCEDURE
ICMCLO	000034	INTEGER*2 VARIABLE
COMMON BLOCK /LINE/	LENGTH 002000	
IBUFF	000000	INTEGER*2 ARRAY (512)

```

RT-11 FORTRAN IV      U01B-08A    UED 18-APR-79 18:12:43      PAGE 001

0001      SUBROUTINE IRPSF
C
C      FUNCTIONAL DESCRIPTION:
C      THIS ROUTINE WILL READ THE CONTAL PSEUDO-COLOR MEMORY
C      AND WRITE IT TO AN IPS PSEUDO-COLOR FILE.
C
C      1) OPEN FILE FOR READ/WRITE
C      2) OPEN CONTAL FOR PROCESSING
C      3) OPEN CONTAL, WRITE FILE
C      4) READ CONTAL, WRITE FILE
C      5) CLOSE CONTAL, CLOSE IPS FILE
C
C      INTEGER NRTYP(2), NAFTYP(2)
C      INTEGER Ibuff(512)
C      COMMON /LINE/ Ibuff
C      DATA NRTYP /3RPS,3RPS/, NAFTYP /'.P','SE'/
C
C      1)
C      2)
C      NRECS = 1
C      NRECSZ = 256
C      IER = IPUOPE(NRECS,NRECSZ,NRTYP,NAFTYP)
C      3) OPEN CONTAL
C      IER=ICHOPE()
C      4) READ CONTAL, WRITE TO IPS PSEUDO-COLOR
C      IER = IRPS (IBUFF)
C      IER = IPUTPS (IBUFF)
C      5) CLOSE IPS FILE, CLOSE CONTAL
C      CALL IPCLOS
0012

```

```

C      IER = ICMCLO
C
0013 C
C
0014 RETURN
0015 END

```

RT-11 FORTRAN IV STORAGE MAP

NAME	OFFSET	ATTRIBUTES
NFTYP	000014	INTEGER*2 ARRAY (2)
NFTYP	000020	INTEGER*2 ARRAY (2)
NRECS	000024	INTEGER*2 VARIABLE
NRECS	000026	INTEGER*2 VARIABLE
IEP	000030	INTEGER*2 VARIABLE
IEP	000030	INTEGER*2 PROCEDURE
ICMPE	000000	INTEGER*2 PROCEDURE
ICMPE	000000	INTEGER*2 PROCEDURE
IPUTPS	000000	INTEGER*2 PROCEDURE
IPUTPS	000000	INTEGER*2 PROCEDURE
ICMCLD	000032	INTEGER*2 VARIABLE
COMMON BLOCK /LINE/	LENGTH	002000
IEUFF	000000	INTEGER*2 ARRAY (512)

PAGE 001

PT-11 FORTRAN IV U018-08A UED 18-APR-79 18:12:46

```

0001 SUBROUTINE IUCMF
C
C      FUNCTIONAL DESCRIPTION:
C      THIS ROUTINE WILL READ AN IPS GRAPHICS (.GRA) FILE
C      AND WRITE IT TO THE CONTAL GRAPHICS MEMORY.
C
C      1) OPEN FILE FOR READ/WRITE
C      2) OPEN CONTAL FOR PROCESSING
C      3) OPEN CONTAL FOR PROCESSING
C      4) READ FILE, WRITE CONTAL
C      5) CLOSE CONTAL, CLOSE IPS FILE
C

```


RT-11 FORTRAN IV STORAGE MAP

NAME	OFFSET	ATTRIBUTES
NAFTYP	000014	INTEGER*2 ARRAY (2)
INNO	000062	INTEGER*2 VARIABLE
IER	000064	INTEGER*2 VARIABLE
IPROPE	000066	INTEGER*2 PROCEDURE
NRECS	000070	INTEGER*2 VARIABLE
ICMOSZ	000070	INTEGER*2 PROCEDURE
ICMOPE	000072	INTEGER*2 VARIABLE
IPRDCM	000072	INTEGER*2 PROCEDURE
ICMOS	000074	INTEGER*2 PROCEDURE
ICMCLC	000074	INTEGER*2 VARIABLE
COMMON BLOCK /LINE/ LENGTH 002000		
IBUFF	000000	INTEGER*2 ARRAY (512)

PAGE 001

RT-11 FORTRAN IV U018-03A UED 18-APR-79 18:12:49

0001 C SUBROUTINE IRGWF

FUNCTIONAL DESCRIPTION:

THIS ROUTINE WILL READ THE CORTAL GRAPHICS MEMORY AND WRITE IT TO AN IPS GRAPHICS (.GRA) FILE.

1) QUERY USER FOR GRAPHICS NUMBER

2) OPEN FILE FOR READ/WRITE

3) OPEN CORTAL FOR PROCESSING

4) READ CORTAL, WRITE FILE

5) CLOSE CORTAL, CLOSE IPS FILE

0002 C INTEGER NAFTYP(2),NAFTYP(2)

0003 C INTEGER IBUFF(512)

0004 C COMMON /LINE/ IBUFF

0005 C DATA NAFTYP /3RGRA,3RGRA/, NAFTYP /'.G','RA'/

1) QUERY USER FOR GRAPHICS NUMBER

```

0006      WRITE(7,110)
0007      FORMAT(' INPUT GRAPHICS NUMBER: ',8)
0008      READ(7,120) IMNO
0009      FORMAT(16)
0010      ,
0011      2)
0012      NRECS = 256
0013      NRECSZ = 16
0014      IER = IPOUPE(NRECS,NRECSZ,NFTYP,NAFTYP)
0015      ,
0016      3) OPEN CONTAL
0017      IER=ICHOPE( )
0018      ,
0019      4) READ CONTAL, WRITE TO IPS GRAPHICS
0020      DO 100 J=1,256
0021      IER = IRGM (IMNO,J,IBUFF)
0022      IER = IPUTGM (IBUFF)
0023      CONTINUE
0024      ,
0025      5) CLOSE IPS FILE, CLOSE CONTAL

```

PAGE 002

RT-11 FORTRAN IV U013-08A UED 18-APR-79 18:12:49

```

0013      CALL IPCLOS
0019      IER = ICNCLO
0020      RETURN
0021      END

```

STORAGE MAP

NAME	OFFSET	ATTRIBUTES
INSTR	000000	INSTR
INSTR2	000000	INSTR2
INSTR3	000000	INSTR3
INSTR4	000000	INSTR4
INSTR5	000000	INSTR5
INSTR6	000000	INSTR6
INSTR7	000000	INSTR7
INSTR8	000000	INSTR8
INSTR9	000000	INSTR9
INSTR10	000000	INSTR10
INSTR11	000000	INSTR11
INSTR12	000000	INSTR12
INSTR13	000000	INSTR13
INSTR14	000000	INSTR14
INSTR15	000000	INSTR15
INSTR16	000000	INSTR16
INSTR17	000000	INSTR17
INSTR18	000000	INSTR18
INSTR19	000000	INSTR19
INSTR20	000000	INSTR20
INSTR21	000000	INSTR21
INSTR22	000000	INSTR22
INSTR23	000000	INSTR23
INSTR24	000000	INSTR24
INSTR25	000000	INSTR25
INSTR26	000000	INSTR26
INSTR27	000000	INSTR27
INSTR28	000000	INSTR28
INSTR29	000000	INSTR29
INSTR30	000000	INSTR30
INSTR31	000000	INSTR31
INSTR32	000000	INSTR32
INSTR33	000000	INSTR33
INSTR34	000000	INSTR34
INSTR35	000000	INSTR35
INSTR36	000000	INSTR36
INSTR37	000000	INSTR37
INSTR38	000000	INSTR38
INSTR39	000000	INSTR39
INSTR40	000000	INSTR40
INSTR41	000000	INSTR41
INSTR42	000000	INSTR42
INSTR43	000000	INSTR43
INSTR44	000000	INSTR44
INSTR45	000000	INSTR45
INSTR46	000000	INSTR46
INSTR47	000000	INSTR47
INSTR48	000000	INSTR48
INSTR49	000000	INSTR49
INSTR50	000000	INSTR50
INSTR51	000000	INSTR51
INSTR52	000000	INSTR52
INSTR53	000000	INSTR53
INSTR54	000000	INSTR54
INSTR55	000000	INSTR55
INSTR56	000000	INSTR56
INSTR57	000000	INSTR57
INSTR58	000000	INSTR58
INSTR59	000000	INSTR59
INSTR60	000000	INSTR60
INSTR61	000000	INSTR61
INSTR62	000000	INSTR62
INSTR63	000000	INSTR63
INSTR64	000000	INSTR64
INSTR65	000000	INSTR65
INSTR66	000000	INSTR66
INSTR67	000000	INSTR67
INSTR68	000000	INSTR68
INSTR69	000000	INSTR69
INSTR70	000000	INSTR70
INSTR71	000000	INSTR71
INSTR72	000000	INSTR72
INSTR73	000000	INSTR73
INSTR74	000000	INSTR74
INSTR75	000000	INSTR75
INSTR76	000000	INSTR76
INSTR77	000000	INSTR77
INSTR78	000000	INSTR78
INSTR79	000000	INSTR79
INSTR80	000000	INSTR80
INSTR81	000000	INSTR81
INSTR82	000000	INSTR82
INSTR83	000000	INSTR83
INSTR84	000000	INSTR84
INSTR85	000000	INSTR85
INSTR86	000000	INSTR86
INSTR87	000000	INSTR87
INSTR88	000000	INSTR88
INSTR89	000000	INSTR89
INSTR90	000000	INSTR90
INSTR91	000000	INSTR91
INSTR92	000000	INSTR92
INSTR93	000000	INSTR93
INSTR94	000000	INSTR94
INSTR95	000000	INSTR95
INSTR96	000000	INSTR96
INSTR97	000000	INSTR97
INSTR98	000000	INSTR98
INSTR99	000000	INSTR99
INSTR100	000000	INSTR100
INSTR101	000000	INSTR101
INSTR102	000000	INSTR102
INSTR103	000000	INSTR103
INSTR104	000000	INSTR104
INSTR105	000000	INSTR105
INSTR106	000000	INSTR106
INSTR107	000000	INSTR107
INSTR108	000000	INSTR108
INSTR109	000000	INSTR109
INSTR110	000000	INSTR110
INSTR111	000000	INSTR111
INSTR112	000000	INSTR112
INSTR113	000000	INSTR113
INSTR114	000000	INSTR114
INSTR115	000000	INSTR115
INSTR116	000	

PT-11 FORTRAN IV U01B-08A WED 18-APR-79 18:12:52 PAGE 001

```

00001 SUBROUTINE IUWFM
      ,      FUNCTIONAL DESCRIPTION:
      ,      THIS ROUTINE WILL READ AN IPS FUNCTION (.TRA ) FILE
      ,      AND WRITE IT TO THE COMTAL FUNCTION MEMORY.
      ,
      ,      1) QUERY USER FOR FUNCTION NUMBER
      ,      2) OPEN FILE FOR READ/WRITE
      ,      3) OPEN COMTAL FOR PROCESSING
      ,      4) READ FILE, WRITE COMTAL
      ,      5) CLOSE COMTAL, CLOSE IPS FILE
      ,
      ,      INTEGER NFTYP(2)
      ,      INTEGER IBUFF(512)
      ,      COMMON /LINE/ IBUFF
      ,
      ,      DATA NFTYP /3RTA,3RTA/
      ,
      ,      1) QUERY USER FOR FUNCTION NUMBER
      ,
      ,      WRITE(7,110)
      ,
00002
00003
00004
00005
00006

```

```

0007 110      FORMAT(' INPUT FUNCTION NUMBER: ',8)
0008 120      READ(7,120) INNO
0009      FORMAT(16)
C
C
C
C
C
C
0010      IER = IPOPE(NRECS,NRECSZ,NFTYP)
C
C
C
0011      IER=ICHOPE()
C
C
C
C
C
C
0012      IER = IPRDFM (IBUFF)
0013      IER = IUFM (INNO,IBUFF)
C
C
C
C
0014      CALL IPCLOS
0015      IER = ICNCLO
C
C
0016      RETURN
0017      END

```

4) READ IPS FILE, WRITE TO CORTAL FUNCTION MEMORY

```

RT-11 FORTRAN IV      STORAGE MAP
NAME  OFFSET  ATTRIBUTES
NFTYP 000014  INTEGER12 ARRAY (2)
INNO  000062  INTEGER12 VARIABLE
IER    000064  INTEGER12 VARIABLE
IPOPE  000066  INTEGER12 PROCEDURE
NRECS  000068  INTEGER12 VARIABLE
NRECSZ 000070  INTEGER12 VARIABLE
ICHOPE 000072  INTEGER12 PROCEDURE
IPRDFM 000074  INTEGER12 PROCEDURE
IUFM   000076  INTEGER12 PROCEDURE
IPCLOS 000078  INTEGER12 PROCEDURE
ICNCLO 000080  INTEGER12 VARIABLE
COMMON BLOCK /LINE/  LENGTH 002000
IBUFF  000000  INTEGER12 ARRAY (512)

```

```

0001 C SUBROUTINE IRFMF
      C
      C FUNCTIONAL DESCRIPTION:
      C
      C THIS ROUTINE WILL READ THE CONTAL FUNCTION MEMORY
      C AND WRITE IT TO AN IPS TRANSFER FUNCTION (.TRA) FILE.
      C
      C 1) QUERY USER FOR FUNCTION NUMBER
      C 2) OPEN FILE FOR READ/WRITE
      C 3) OPEN CONTAL FOR PROCESSING
      C 4) READ CONTAL, WRITE FILE
      C 5) CLOSE CONTAL, CLOSE IPS FILE
      C
      C
      C INTEGER NFTYP(2),NAFTYP(2)
      C INTEGER Ibuff(512)
      C COMMON /LINE/ Ibuff
      C DATA NFTYP /3RTA,3RTA/, NAFTYP /'.T','RA'/
      C
      C 1) QUERY USER FOR FUNCTION NUMBER
      C
      C WRITE(7,110)
      C FORMAT('INPUT FUNCTION NUMBER: ',9)
      C READ(7,120) INNO
      C FORMAT(I6)
      C
      C 2)
      C
      C NRECS = 1
      C NRECSZ = 128
      C IER = IPDOPE(NRECS,NRECSZ,NFTYP,NAFTYP)
      C
      C 3) OPEN CONTAL
      C
      C IER=ICMOPE()
      C
      C 4) READ CONTAL, WRITE TO IPS TRANSFER FUNCTION FILE
      C
      C IEP = IRFM (INNO,IBUFF)
      C IER = IPUTFM (IBUFF)

```

```

C      5) CLOSE IPS FILE, CLOSE COMTEL
C
0016 C      CALL IPCLOS
0017 C      IER = ICNCLO
C

```

PAGE 002

U018-08A WED 18-APR-79 18:12:54

```

RT-11 FORTRAN IV
C
0018 C      RETURN
0019 C      END

```

```

RT-11 FORTRAN IV      STORAGE MAP
NAME  OFFSET  ATTRIBUTES
NFTYP 000014 INTEGER12 ARRAY (2)
NAFTYP 000020 INTEGER12 ARRAY (2)
IMNO 000056 INTEGER12 VARIABLE
NRECS 000070 INTEGER12 VARIABLE
NRECS2 000072 INTEGER12 VARIABLE
IER 000074 INTEGER12 VARIABLE
IPUOPE 000080 INTEGER12 PROCEDURE
ICMOPE 000088 INTEGER12 PROCEDURE
IPUFM 000096 INTEGER12 PROCEDURE
IPCLOS 000098 INTEGER12 PROCEDURE
ICMLO 000076 INTEGER12 VARIABLE
COMMON BLOCK /LINE/  LENGTH 002000
IEUFF 000000 INTEGER12 ARRAY (512)

```

PAGE 001

U018-08A WED 18-APR-79 18:12:57

```

RT-11 FORTRAN IV

```

```

0001 SUBROUTINE IUCBF
      THIS ROUTINE WILL SEND ONE LINE OF TEXT
      TO THE CORTAL COMMAND BUFFER
      1) QUERY USER FOR TEXT
      2) OPEN CORTAL
      3) SEND TO CORTAL
      4) CLOSE CORTAL

0002 LOGICALS: ITEXT(132)
0003 COMMON /LINE/ ITEXT

      1) QUERY USER
0004 WRITE(7,100)
0005 FORMAT(' ENTER ONE LINE OF COMMAND TEXT, TERMINATE WITH ',
      & '8, CR. ',/,)
0006 READ (7,200)L, (ITEXT(J),J=1,L)
0007 FORMAT(0,130a1)

      2) OPEN CORTAL
0008 IER=ICNOPE(1)

      3) SEND TO CORTAL
0009 IER = IUCB(ITEXT)

      4) CLOSE CORTAL
0010 IER = ICNCLO
0011 RETURN
0012 END

```

RT-1: FORTRAN IV STORAGE MAP
NAME OFFSET ATTRIBUTES

```

1 000120 INTEGER12 VARIABLE
1 000122 INTEGER12 VARIABLE
1 000124 INTEGER12 VARIABLE
1 000000 INTEGER12 PROCEDURE
1 000000 INTEGER12 PROCEDURE
1 000126 INTEGER12 VARIABLE
COMMON BLOCK LINE LENGTH 000204
ITEXT 000000 LOGICAL11 ARRAY (132)
1

```



```

      FILE NAME: CPIPLB.FOR
      DATE LAST EDITED: 21-MAR-79

      HISTORY:

      DATE:      MODIFICATION
      ----      -
      6-MAR-79   INITIAL WRITING
      7-MAR-79   ADD MORE ROUTINES
      8-MAR-79   FIX BUGS
      9-MAR-79   FIX MORE BUGS
      19-MAR-79  CHANGE SUBS TO FUNCTIONS
      21-MAR-79  ADD MORE FUNCTIONS
  
```

THIS FILE CONTAINS THE GENERAL SUBROUTINES
FOR THE COMTAL READ AND WRITE IPS FILES PROGRAM.

INTEGER FUNCTION IPROPE(IRECS,IRECSZ,IFTYP)

THIS ROUTINE OPENS AN IPS FILE FOR READING.

```

      IREC = NUMBER OF RECORDS IN FILE
      IRECSZ = NUMBER OF WORDS PER RECORD
      IFTYP = RAD50 FILE TYPE
  
```

INTEGER IFTYP(2)

DATA DEFINITION

INTEGER INFL(4),OUTFL(4),EXT(2),FSPEC(39),BUFF(256,2)

EQUIVALENCE (INFL,FSPEC(16)),(OUTFL,FSPEC(16))

COMMON /FILES/ FSPEC,EXT,NRECS,NRECSZ,ICHIPS,
IACT,NABREC,BUFF

```

0006 DATA IACT / 1 /
0007 DATA DATA EXT /3RIMG,3RING/
0008 END OF DATA DEFINITION
0009 *****
0010
0011 . 1) QUERY USER FOR INPUT FILE NAME

```

RT-11 FORTRAN IV U01B-08A UED 21-MAR-79 17:21:08 PAGE 002

```

0000 ; 2) GET CHANNEL
0001 ; 3) OPEN FILE
0002 ; 4) READ HEADER BLOCK, FILL IN PARAMETERS

```

```

0003 . 1) QUERY USER
0004 IPROPE = 0
0005 PAUSE ' IPROPE 1, BEGIN '
0006
0007 90 WRITE(7,100)
0008 100 FORMAT(' ENTER INPUT FILE NAME ','')
0009 IF(ICS1(FSPEC,IFTYP,...0).NE.0) GO TO 90

```

```

0010 124 WRITE(7,124)
0011 124 FORMAT(' ', FSPEC , ' ')
0012 123 WRITE(7,123)FSPEC
0013 123 FORMAT(' ',1007)

```

```

0014 . 2) GET CHANNEL
0015 PAUSE 'IPROPE 3 JUST PRIOR TO GETC '
0016
0017 ICHIPS=IGETC()
0018 IF(ICHIPS.LT.0) STOP 'BAD CHANNEL'

```

```

0019 . 3) OPEN FILE

```

```

0015      PAUSE / IPROPE 4 JUST PRIOR TO LOOKUP /
0016      INSIZ=LOOKUP(ICHIPS,INFL)
0017      IF(INSIZ.LT.0) GO TO 1100
0018
0019      4) READ HEADER
0020      PAUSE / IPROPE 5 JUST PRIOR TO READU /
0021      NABREC=0
0022      IC=IREADU(256,BUFF(1,1),NABREC,ICHIPS)
0023      IF(IC.LT.0) GO TO 1100
0024
0025      4.5) FILL HEADER INFORMATION
0026
0027
0028
0029
0030
0031
0032
0033
0034
0035

```

PAGE 003

```

RT-11 FORTRAN IV      0018-08A      UED 21-MAR-79 17:21:08
0023      NRECS = BUFF(7,1)
0024      NRECSZ = BUFF(4,1)
0025      CHANGE RECORD SIZE FROM BYTES TO WORDS
0026      NRECSZ = (NRECSZ+1) / 2
0027      IRECS = NRECS
0028      IRECSZ = NRECSZ
0029      SET FLAG BUFFER EMPTY
0030      IACT = 1
0031      NABREC = NABREC + 1
0032      RETURN
0033      ERROR OCCURED
0034      WRITE(7,110)
0035      FORMAT(' ERROR , IPROPE ')
0036      IPROPE = -1
0037      RETURN
0038      END

```

RT-11 FORTRAN IV STORAGE MAP

NAME	OFFSET	ATTRIBUTES
...

NAME	TYPE	VALUE	ARRAY (2)
IFYP	000000	INTEGER12	PARAMETER
IPROPE	000002	INTEGER12	VARIABLE
ISECS	000014	INTEGER12	PARAMETER
ISECS=	000016	INTEGER12	PARAMETER
IS1	000000	INTEGER12	PROCEDURE
IS1C	000000	INTEGER12	PROCEDURE
IS1C=	000132	INTEGER12	VARIABLE
LOOKUP	000000	INTEGER12	PROCEDURE
IC	000134	INTEGER12	VARIABLE
IC=	000000	INTEGER12	PROCEDURE
IRADW	000000	INTEGER12	PROCEDURE

COMMON BLOCK /IFILES/ LENGTH 002134

```

FSPC      000000  INTEGER12  ARRAY (39)
EXT       000116  INTEGER12  ARRAY (2)
NRECS     000122  VARIABLE
NCHPS     000124  VARIABLE
ICHP5     000126  VARIABLE
IACT      000130  VARIABLE
NAPREC    000132  VARIABLE
BUFF      000134  INTEGER12  ARRAY (256.2) VECTORED
INFL      000036  INFL
OUTFL     000036  INTEGER12  ARRAY (4)

```

PT-11 FORTRAN IV V01B-08A WED 21-MAR-79 17:21:11 PAGE 001

```
00001 INTEGER FUNCTION IPNOPE(IRECS,IRECSZ,IFTYP,IAFTYP)
```

THIS ROUTINE OPENS AN IPS FILE FOR OUTPUT.

```
IRECS      •  NUMBER OF RECORDS IN FILE
IRECSZ     •  RECORD SIZE IN WORDS
IFTYP      •  RAD50 FILE TYPE
IAFTYP     •  ASCII FILE TYPE
```

00022
INTEGER IFTYP(2), IAFYP(2)


```

0013 C      BUFF(7,1) = IRECS
0014 C      BUFF(4,1) = IRECSZ
0015 C      BUFF(2,1) = IFTYP(1)
0016 C      BUFF(3,1) = IFTYP(2)
0017 C      ,
0018 C      ,      SINCE NUMBER OF RECORDS = 256
0019 C      ,      NUMBER OF BLOCKS = 256 * (256/512) + 1 , = 129
0020 C      ,
0021 C      ARECS=IRECS
0022 C      ARECSZ=IRECSZ
0023 C      ASIZ = ((ARECS*ARECSZ)/256.) + 2.
0024 C      BUFF(5,1) = ASIZ
0025 C      ,
0026 C      2) GET CHANNEL
0027 C      ,
0028 C      ICHIPS=IGETC()
0029 C      IF(ICHIPS.LT.0) STOP 'BAD CHANNEL'
0030 C      ,
0031 C      3) QUERY USER
0032 C      ,
0033 C      WRITE(7,100)
0034 C      FORMAT(' ENTER OUTPUT FILE NAME ',/)
0035 C      IF(ICS1(FSPEC,IFTYP,,,0).NE.0) GO TO 90
0036 C      ,
0037 C      4) OPEN FILE
0038 C      ,
0039 C      IF(IENTER(ICHIPS,OUTFL,0).LT.0) GO TO 1100
0040 C      ,
0041 C      5) WRITE HEADER
0042 C      ,
0043 C      NABREC=0
0044 C      IC= IURITU(256,BUFF(1,1),NABREC,ICHIPS)
0045 C      ,
0046 C      NABREC = NABREC+1
0047 C      RETURN

```

PT-11 FORTRAN IV UED 21-MAR-79 17:21:11

U018-084

PT-11 FORTRAN IV

```

      C
      C
      C      ERROR OCCURED
      C
      0034 1100 WRITE(7,110)
      0035 110  FORMAT(' ERROR , IPUOPE ')
      C
      0036 IPUOPE = -1
      0037 RETURN
      0038 END
  
```

PT-11 FORTRAN IV STORAGE MAP

NAME	OFFSET	ATTRIBUTES
IFTYP	000020	INTEGER12 PARAMETER ARRAY (2)
IAFTYP	000022	INTEGER12 PARAMETER ARRAY (2)
IAEXT	000024	INTEGER12 ARRAY (2)
IPUOPE	000030	INTEGER12 VARIABLE
IRECS	000014	INTEGER12 PARAMETER VARIABLE
IRECS2	000016	INTEGER12 PARAMETER VARIABLE
J	000142	INTEGER12 VARIABLE
ARECS	000144	REAL14 VARIABLE
ARECS2	000150	REAL14 VARIABLE
AS22	000154	REAL14 VARIABLE
IGETC	000000	INTEGER12 PROCEDURE
IGSI	000000	INTEGER12 PROCEDURE
IENTER	000000	INTEGER12 PROCEDURE
IG	000160	INTEGER12 VARIABLE
IURITU	000000	INTEGER12 PROCEDURE
COMMON BLOCK /FILES/ LENGTH 002134		
FSPEC	000000	INTEGER12 ARRAY (30)
EXT	000116	INTEGER12 ARRAY (2)
NRECS	000122	INTEGER12 VARIABLE
NRECS2	000124	INTEGER12 VARIABLE
ICHIPS	000126	INTEGER12 VARIABLE
IMCT	000130	INTEGER12 VARIABLE
NABPEC	000132	INTEGER12 VARIABLE
BUFF	000134	INTEGER12 ARRAY (256,2) VECTORED
INEL	000036	INTEGER12 ARRAY (4)
OUTFL	000036	INTEGER12 ARRAY (4)

INTEGER FUNCTION IPRDIN(IBUFF)

THIS ROUTINE WILL READ ONE RECORD FROM AN IPS FILE
AND RETURN THE RECORD TO THE USER BUFFER. THE IPS
FILE MUST HAVE BEEN ALREADY OPENED BY A CALL TO
SUBROUTINE IPROPE.

PARAMETERS:

IBUFF - INTEGER ARRAY FOR THE RECORD

XX

DATA DEFINITION

XX

THIS PART OF THE DATA DEF. APPLIES ONLY TO THIS ROUTINE

INTEGER RECST, RECPOS

LOGICAL I1 IBUFF(1), BUFF(512,2)

NOTE: IBUFF SHOULD BE MADE INTEGER TO SPEED UP DO
LOOPS, CANT UNTIL BYTE SWAP FIXED

XX

INTEGER INFL(4), OUTFL(4), EXT(2), FSPEC(39)

EQUIVALENCE (INFL, FSPEC(16)), (OUTFL, FSPEC(16))

COMMON /IFILES/ FSPEC, EXT, NRECS, NRECSZ, ICHIPS,
IACT, NABREC, BUFF

XX

DATA IBFSZ /256 /

(----- SWAP -----)

END OF DATA DEFINITION

1) CHECK FOR CONDITION OF THE READ BUFFER
THERE ARE 3 POSSIBLE CONDITIONS.

- 1) BUFFER IS EMPTY (WHEN FIRST CALLED)
- 2) FIRST BUFFER IS ACTIVE
- 3) SECOND BUFFER IS ACTIVE

2) READ RECORD INTO USER BUFFER STARTING FROM THE
POSITION DEFINED BY IACT, AND RECS. IACT IS THE
ACTIVE BUFFER NUMBER (1,2,3), AND RECS IS THE START
OF THE RECORD IN THE BUFFER.

NOTE: THERE ARE ONLY 2 POSSIBLE ACTIVE BUFFERS, 2 AND 3.
ACTIVE BUFFER = 1 MEANS BOTH BUFFERS ARE EMPTY.

3) IF THE END OF A BUFFER OCCURS DURING A READ, THE
ACTIVE BUFFER WILL BE CHANGED, AND A NEW BLOCK READ
INTO THE INACTIVE BUFFER (THE ONE JUST ENDED).

1) DETERMINE STATE OF BUFFER

PAUSE ' IPREAD 1 - BEGIN '

IPRDIM = 0

GO TO (100,200,300),IACT

2.1) BOTH BUFFERS ARE EMPTY READ 2 BLOCKS,
ACTIVE BUFFER = 1

SET ACTIVE BUFFER = BUFFER # 1, (IACT-2)

0002

0003

```

0010 100 IACT=2
      .      READ 2 BLOCKS INTO THE BUFFERS
0011      IC=IREADU(S12,BUFF(1,1),NABREC,ICHIPS)
      .      INCREMENT ABSOLUTE BLOCK NUMBER
      .      AND THEN READ RECORD
0012      NABREC = NABREC+2
0013      RECST=0
0014      GO TO 10

```

RT-11 FORTRAN IV U018-08A UED 21-MAR-79 17:21:14 PAGE 003

```

      .      2.2) READ RECORD INTO USER BUFFER - ACT BUFFER = 2
      .      (IACT = 2, BUT FIRST BUFFER)

```

```

0015      DO 2000 J=1,NRECSZ
0016      RECPOS = RECST+J
0017      IF (RECPOS.GT.1BF5Z) GO TO 2100
      .      (----- SUAP -----
0019      NJ = 2xJ
0020      MJ = 2x(RECPOS)
0021      Ibuff(NJ-1) = BUFF(MJ,1)
0022      Ibuff(NJ) = BUFF(MJ-1,1)
      .      (----- SUAP -----

```

```

0023      CONTINUE
0024      RECST = RECST + NRECSZ
0025      GO TO 5000

```

```

      .      END OF BUFFER BEFORE END OF RECORD
      .      SWITCH BUFFERS, FINISH READING RECORD

```

```

0026      IACT=3
0027      RECPOS=0

```

```

C      DO 2300 N-J,NRECSZ
C      RECPOS=RECPOS + 1
C
C      C----- SUAP -----
C      NJ = 28J
C      NJ = 28(RECPOS)
C      IBUFF(NJ-1) = BUFF(NJ,2)
C      IBUFF(NJ) = BUFF(NJ-1,2)
C      C----- SUAP -----
C
C      CONTINUE
C      RECST=RECPOS
C
C      4.1) READ BLOCK INTO EMPTY FIRST BUFFER
C      IC=IREADN(256,BUFF(1,1),NABREC,ICHIPS)

```

1-38

```

RT-11 FORTRAN IV      U01B-08A      UED 21-MAR-79 17:21:14      PAGE 004
0037      NABREC=NABREC+1
0038      GO TO 5000
C
C      2.3) READ RECORD INTO USER BUFFER - ACTIVE BUFFER = 3
C
0039      DO 3000 J=1,NRECSZ
0040      RECPOS =RECST + J
0041      IF(RECPOS.GT.1BFSZ) GO TO 3100
C
C      C----- SUAP -----
C      NJ = 28J
C      NJ = 28(RECPOS)
C      IBUFF(NJ-1) = BUFF(NJ,2)
C      IBUFF(NJ) = BUFF(NJ-1,2)
C      C----- SUAP -----
C
0047      3000 CONTINUE
0048      RECST=RECST + NRECSZ
0049      GO TO 5000

```

1-39


```

000130 INTEGER12 VARIABLE
000132 INTEGER12 VARIABLE
000134 LOGICAL*1 ARRAY (512,2) VECTORED
000136 INTEGER12 ARRAY (4)
000138 OUTFL

```

RT-11 FORTRAN IV U01B-08A WED 21-MAR-79 17:21:19 PAGE 001

INTEGER FUNCTION IPUTIM (IBUFF)

```

* THIS ROUTINE WILL WRITE ONE RECORD TO AN IPS FILE
* FROM THE RECORD IN THE USER BUFFER. THE IPS
* FILE MUST HAVE BEEN ALREADY OPENED BY A CALL TO
* SUBROUTINE 'IPUOPE'.

```

PARAMETERS:

IBUFF - INTEGER ARRAY FOR THE RECORD

DATA DEFINITION

THIS PART OF THE DATA DEF. APPLIES ONLY TO THIS ROUTINE

<----- SUAP -----

LOGICAL*1 IBUFF(1), BUFF(256,4)

NOTE: BUFF IS CHANGED TO 128,4 FROM 256,2 FOR THIS ROUTINE

<----- SUAP -----

INTEGER INFL(4),OUTFL(4),EXT(2),FSPEC(39)

EQUIVALENCE (INFL,FSPEC(16)),(OUTFL,FSPEC(16))

```

0000      <----- SUAP -----
0001      COMMON /FILES/ FSPEC,EXT,NRECS,NRECSZ,ICHIPS,
0002      IACT,NABREC,BUFF
0003      !
0004      !
0005      !
0006      THIS PART OF THE DATA DEF. APPLIES ONLY TO THIS ROUTINE
0007      DATA N / 1 /

```

PT-11 FORTRAN IV V018-08A UED 21-MAR-79 17:21:19 PAGE 002

END OF DATA DEFINITION

```

1) FILL BUFF WITH USER DATA
2) UPON 4 DATA RECORDS, WRITE ONE BLOCK

```

```

1) FILL BUFFER WITH DATA

```

PAUSE / IPWRITE 1 BEGIN /

NOTE:

THE CODE MARKED <-----SUAP-----
IS FOR THE BYTE SUAP THAT THE COMTAL
IS SUPPOSED TO DO, BUT AT THIS TIME
DOESN'T.

IPUTIM = 0

```

0005 DO 100 J=1,128
0006     , <----- SUAP -----
0007 MJ = J*2
0008     , <----- SUAP -----
0009
0010 BUFF(MJ,N) = IBUFF(MJ-1)
0011 BUFF(MJ-1,N) = IBUFF(MJ)
0012 CONTINUE
0013 N = N+1
0014 IF(N.GT.4) GO TO 1000
0015 RETURN
0016     , 2) WRITE OUT BUFFER
0017     ,

```

PAGE 003

```

PT-11 FORTRAN IV      U018-08A   UED 21-MAR-79 17:21:19
0017 1000 N = 1
0018 C
0019 IC = IURITU(512,BUFF(1,1),NABREC,ICHIPS)
0020 NABREC = NABREC + 2
0021 RETURN
0022 END

```

PT-11 FORTRAN IV STORAGE MAP

NAME	OFFSET	ATTRIBUTES
IBUFF	000014	LOGICAL11 PARAMETER ARRAY (1)
IPUTIM	000016	INTEGER12 VARIABLE
I	000020	INTEGER12 VARIABLE
J	000034	INTEGER12 VARIABLE
MJ	000036	INTEGER12 VARIABLE
IC	000040	INTEGER12 VARIABLE


```

TURITU 000000 INTEGER12 PROCEDURE
COMMON BLOCK IFILES/ LENGTH 002134
FSPEC 000000 INTEGER12 ARRAY (39)
EXT 000116 INTEGER12 ARRAY (2)
NRECS 000122 INTEGER12 VARIABLE
ICHIPS 000124 INTEGER12 VARIABLE
NABREC 000126 INTEGER12 VARIABLE
INFL 000130 INTEGER12 VARIABLE
OUTFL 000132 LOGICAL11 ARRAY (256,4) VECTORED
INFL 000134 INTEGER12 ARRAY (4)
OUTFL 000036 INTEGER12 ARRAY (4)

```

PT-11 FORTRAN IV U01B-08A WED 21-MAR-79 17:21:22 PAGE 001

```

0001 C SUBROUTINE IPCLOS
C , THIS ROUTINE WILL CLOSE THE IPS FILE
C
C DATA DEFINITION
C
0002 C INTEGER INFL(4),OUTFL(4),EXT(2),FSPEC(39),BUFF(256,2)
0003 C EQUIVALENCE (INFL,FSPEC(16)),(OUTFL,FSPEC(16))
0004 C COMMON /IFILES/ FSPEC,EXT,NRECS,NRECSZ,ICHIPS,
C IACT,NABREC,BUFF
C
C END OF DATA DEFINITION
C*****
0005 C CALL CLOSEC(ICHIPS)
0006 C RETURN
0007 C END

```

RT-11 FORTRAN IV STORAGE MAP

NAME	OFFSET	ATTRIBUTES
CLOSEC	000000	REAL14 PROCEDURE
COMMON BLOCK	IFILES/	LENGTH 002134
FSPEC	000000	INTEGER12 ARRAY (39)
EXT	000116	INTEGER12 ARRAY (2)
NRECS	000122	INTEGER12 VARIABLE
NRECSZ	000124	INTEGER12 VARIABLE
ICMIPS	000126	INTEGER12 VARIABLE
IAC	000130	INTEGER12 VARIABLE
NABREC	000132	INTEGER12 VARIABLE
IBUFF	000134	INTEGER12 ARRAY (256,2) VECTORED
INFL	000036	INTEGER12 ARRAY (4)
OUTFL	000036	INTEGER12 ARRAY (4)

RT-11	FORTTRAN IL	U01B-08A	VED 21-MAR-79	17:21:24	PAGE 001
0001	C	INTEGER FUNCTION IPRDPS (IBUFF)			
0002	C	FUNCTIONAL DESCRIPTION:			
0003	C	THIS ROUTINE READS ONE PSEUDO COLOR MEMORY			
0004	C	RECORD INTO THE USER BUFFER			
0005	C	THIS APPLIES ONLY TO THIS ROUTINE:			
0006	C	INTEGER IBUFF(1)			
0007	C	*****			
0008	C	DATA DEFINITION			
0009	C	INTEGER INFL(4),OUTFL(4),EXT(2),FSPEC(39),BUFF(256,2)			
0010	C	EQUVALENCE (INFL,FSPEC(16)),(OUTFL,FSPEC(16))			
0011	C	COMMON /IFILES/ FSPEC,EXT,NRECS,NRECSZ,ICMIPS,			
0012	C	IAC,NABREC,BUFF			
0013	C	E D OF DATA DEFINITION			
0014	C	*****			

```

C
0006      IPADPS = 0
C
0007      NSIZ = 256
0008      NR = 1
C
0009      IC = IREADU(NSIZ,IBUFF,NR,ICHIPS)
0010      IF(IC.LT.0) IPADPS = IC
C
0012      RETURN
0013      END

```

PT-11 FORTRAN IV STORAGE MAP

NAME	OFFSET	ATTRIBUTES
IBUFF	000014	INTEGER12 PARAMETER ARRAY (1)
IPADPS	000016	INTEGER12 VARIABLE
NSIZ	000024	INTEGER12 VARIABLE
NR	000026	INTEGER12 VARIABLE
IC	000030	INTEGER12 VARIABLE
IREADU	000000	INTEGER12 PROCEDURE
COMMON BLOCK /FILES/ LENGTH 002134		
EXPEC	000000	INTEGER12 ARRAY (39)
ERT	000116	INTEGER12 ARRAY (2)
APPCS	000122	INTEGER12 VARIABLE
NRECS	000124	INTEGER12 VARIABLE
ICHIPS	000126	INTEGER12 VARIABLE
ICHT	000130	INTEGER12 VARIABLE
NREDEC	000132	INTEGER12 VARIABLE
IBUFF	000134	INTEGER12 ARRAY (256,2) VECTORED
INFL	000036	INTEGER12 ARRAY (4)
OUTFL	000036	INTEGER12 ARRAY (4)

PT-11 FORTRAN IV U01B-08A UED 21-MAR-79 17:21:27 PAGE 001

```

C
0001      INTEGER FUNCTION IPUTPS (IBUFF)
C
C      , FUNCTIONAL DESCRIPTION:
C
C      , THIS ROUTINE WRITES THE PSEUDO COLOR MEMORY

```

```

      RECORD TO THE IPS FILE.
      THIS APPLIES ONLY TO THIS ROUTINE:
      INTEGER IBUFF(1)
      *****
      DATA DEFINITION
      INTEGER INFL(4),OUTFL(4),EXT(2),FSPEC(39),BUFF(256,2)
      EQUIVALENCE (INFL,FSPEC(16)),(OUTFL,FSPEC(16))
      COMMON /IFILES/ FSPEC,EXT,NRECS,NRECSZ,ICHIPS,
      IACT,NABREC,BUFF
      END OF DATA DEFINITION
      *****
      IPUTS = 0
      NR=1
      NSIZ=256
      IC = IURITU(NSIZ,IBUFF,NR,ICHIPS)
      IF(IC.LT.0) IPUTS = IC
      RETURN
      END

```

```

RT-11 FORTRAN IV      STORAGE MAP
NAME  OFFSET  ATTRIBUTES
IBUFF 000014  INTEGER*2  PARAMETER  ARRAY (1)
IPUTS 000016  INTEGER*2  VARIABLE
NR     000024  INTEGER*2  VARIABLE
NSIZ   000026  INTEGER*2  VARIABLE
IC     000030  INTEGER*2  VARIABLE
IURITU 000000  INTEGER*2  PROCEDURE
COMMON BLOCK /IFILES/  LENGTH 002134
FSPEC  000000  INTEGER*2  ARRAY (39)

```

```

EXT      000116 INTEGER2 ARRAY (2)
NRECS2   000122 INTEGER2 VARIABLE
NRECS2   000124 INTEGER2 VARIABLE
ICHIPS   000126 INTEGER2 VARIABLE
IACT     000130 INTEGER2 VARIABLE
NABREC   000132 INTEGER2 VARIABLE
BUFF     000134 INTEGER2 ARRAY (256,2) VECTORED
INFL     000036 INTEGER2 ARRAY (4)
OUTFL    000036 INTEGER2 ARRAY (4)

```

RT-11 FORTRAN IV U01B-08A UED 21-MAR-79 17:21:30 PAGE 001

```

0001      INTEGER FUNCTION IPRDGM (IBUFF)
          , FUNCTIONAL DESCRIPTION:
          , THIS ROUTINE READS ONE LINE OF GRAPHICS MEMORY
          , RECORD INTO THE USER BUFFER
          , THIS APPLIES ONLY TO THIS ROUTINE:

          INTEGER IBUFF(1)

          NOTE: BUFF IS DEFINED AS 16 BY 16 FOR 16 ROWS
          16 WORDS (2 BYTES) OF GRAPHICS MEMORY PER BLOCK
          THIS DIFFERS FROM THE OTHERS ROUTINES

          INTEGER BUFF(16,16)
          , *****

          DATA DEFINITION
          INTEGER INFL(4),OUTFL(4),EXT(2),FSPEC(39)
          EQUIVALENCE (INFL,FSPEC(16)),(OUTFL,FSPEC(16))
          COMMON /FILES/ FSPEC,EXT,NRECS,NRECS2,ICHIPS,
          , IACT,NABREC,BUFF
          DATA IRELRC /1/
          END OF DATA DEFINITION
          C*****

```

```

0005      IPRDGH = 0
0006      NSIZ = 256
0007
0008      1) CHECK FOR FIRST CALL - NEED TO READ IN BLOCK
0009      IF(IACT.EQ.1) GO TO 100
0010
0011      2) CHECK FOR BLOCK EMPTY - NEED TO READ IN BLOCK
0012      IF(IRELRC.GE.17) GO TO 100
0013
0014      3) MOVE A RECORD TO USER BUFFER
0015      DO 300 J=1,16

```

PAGE 002

```

RT-11 FORTRAN IV      U018-08A      UED 21-MAR-79 17:21:30
0015      Ibuff(J) = BUFF(J,IRELRC)
0016      300 CONTINUE
0017
0018      4) INCREMENT RELATIVE RECORD NUMBER
0019      IRELRC = IRELRC + 1
0020      RETURN
0021
0022      5) END OF BLOCK READ IN A NEW BLOCK
0023      IACT = 2
0024      IC = IREADU(NSIZ,BUFF(1,1),NABREC,ICHIPS)
0025      IF(IC.LT.0) IPRDGH = IC
0026      IRELRC = 1
0027      NABREC = NABREC + 1
0028      GO TO 200
0029      END

```

RT-11 FORTRAN IV STORAGE MAP

[illegible]

	PARAMETER	ARRAY (1
000014	INTEGER22	
000016	INTEGER22	
000020	INTEGER22	
000024	INTEGER22	
000028	INTEGER22	
000032	INTEGER22	
000036	INTEGER22	
000040	INTEGER22	
000044	INTEGER22	
000048	INTEGER22	
000052	INTEGER22	
000056	INTEGER22	
000060	INTEGER22	
000064	INTEGER22	
000068	INTEGER22	
000072	INTEGER22	
000076	INTEGER22	
000080	INTEGER22	
000084	INTEGER22	
000088	INTEGER22	
000092	INTEGER22	
000096	INTEGER22	
000100	INTEGER22	
000104	INTEGER22	
000108	INTEGER22	
000112	INTEGER22	
000116	INTEGER22	
000120	INTEGER22	
000124	INTEGER22	
000128	INTEGER22	
000132	INTEGER22	
000136	INTEGER22	
000140	INTEGER22	
000144	INTEGER22	
000148	INTEGER22	
000152	INTEGER22	
000156	INTEGER22	
000160	INTEGER22	
000164	INTEGER22	
000168	INTEGER22	
000172	INTEGER22	
000176	INTEGER22	
000180	INTEGER22	
000184	INTEGER22	
000188	INTEGER22	
000192	INTEGER22	
000196	INTEGER22	
000200	INTEGER22	
000204	INTEGER22	
000208	INTEGER22	
000212	INTEGER22	
000216	INTEGER22	
000220	INTEGER22	
000224	INTEGER22	
000228	INTEGER22	
000232	INTEGER22	
000236	INTEGER22	
000240	INTEGER22	
000244	INTEGER22	
000248	INTEGER22	
000252	INTEGER22	
000256	INTEGER22	
000260	INTEGER22	
000264	INTEGER22	
000268	INTEGER22	
000272	INTEGER22	
000276	INTEGER22	
000280	INTEGER22	
000284	INTEGER22	
000288	INTEGER22	
000292	INTEGER22	
000296	INTEGER22	
000300	INTEGER22	
000304	INTEGER22	
000308	INTEGER22	
000312	INTEGER22	
000316	INTEGER22	
000320	INTEGER22	
000324	INTEGER22	
000328	INTEGER22	
000332	INTEGER22	
000336	INTEGER22	
000340	INTEGER22	
000344	INTEGER22	
000348	INTEGER22	
000352	INTEGER22	
000356	INTEGER22	
000360	INTEGER22	
000364	INTEGER22	
000368	INTEGER22	
000372	INTEGER22	
000376	INTEGER22	
000380	INTEGER22	
000384	INTEGER22	
000388	INTEGER22	
000392	INTEGER22	
000396	INTEGER22	
000400	INTEGER22	
000404	INTEGER22	
000408	INTEGER22	
000412	INTEGER22	
000416	INTEGER22	
000420	INTEGER22	
000424	INTEGER22	
000428	INTEGER22	
000432	INTEGER22	
000436	INTEGER22	
000440	INTEGER22	
000444	INTEGER22	
000448	INTEGER22	
000452	INTEGER22	
000456	INTEGER22	
000460	INTEGER22	
000464	INTEGER22	
000468	INTEGER22	
000472	INTEGER22	
000476	INTEGER22	
000480	INTEGER22	
000484	INTEGER22	
000488	INTEGER22	
000492	INTEGER22	
000496	INTEGER22	
000500	INTEGER22	

COMMON BLOCK / IFILES/ LENGTH 001134

```

FDEC      000000      INTEGER12      ARRAY (39)
EXT       000116      INTEGER12      ARRAY (2)
NREGS     000122      INTEGER12      VARIABLE
NREGS2    000124      INTEGER12      VARIABLE
ICMPS     000126      INTEGER12      VARIABLE
IACT      000130      INTEGER12      VARIABLE
NABREC    000132      INTEGER12      VARIABLE
BUFF      000134      INTEGER12      ARRAY (4)
INFL      000036      INTEGER12      ARRAY (4)
OUTFL     000036      INTEGER12      ARRAY (4)
VECTORED  000036      INTEGER12      ARRAY (16,16)

```

PT-11 FORTRAN IV U01B-08A UED 21-MAR-79 17:21:33 PAGE 001

```

0001      INTEGER FUNCTION IPUTCH (IBUFF)
0002      ,      FUNCTIONAL DESCRIPTION:
0003      ,      THIS ROUTINE WRITES THE GRAPHICS MEMORY
0004      ,      RECORD TO THE IPS FILE.
0005      ,      THIS APPLIES ONLY TO THIS ROUTINE:
0006      INTEGER IBUFF(1)

```

```

0003      NOTE: BUFF IS DEFINED AS 16 BY 16 FOR 16 ROUTS
0004      16 WORDS (2 BYTES) OF GRAPHICS MEMORY PER BLOCK
0005      THIS DIFFERS FROM THE OTHERS ROUTINES
0006      INTEGER BUFF(16,16)
0007

```

```

. *****
DATA DEFINITION
INTEGER INFL(4),OUTFL(4),EXT(2),FSPEC(39)
EQUVALENCE (INFL,FSPEC(16)),(OUTFL,FSPEC(16))
COMMON /FILES/ FSPEC,EXT,NRECS,NRECSZ,ICHIPS,
               IACT,NABREC,BUFF
DATA IRELRC /1/
END OF DATA DEFINITION
*****
0008      IPUTGM = 0
0009      NSIZ=256
0010      DO 100      J=1,16
0011      BUFF(J,IRELRC) = IBUFF(J)
0012 100      CONTINUE
0013      IRELRC=IRELRC + 1
0014      IF (IRELRC.GE.17) GO TO 1000
0016      RETURN
C

```

PAGE 002

```

PT-11 FORTRAN IV      U01B-08A      WED 21-MAR-79 17:21:33
C
0017 1000 IRELRC = 1
0018      IC = IURITU(NSIZ,BUFF(1,1),NABREC,ICHIPS)
0019      IF (IC.LT.0) IPUTGM = IC
0021      NABREC =NABREC + 1
0022      RETURN
0023      END

```


RT-11 FORTRAN IV STORAGE MAP

NAME	OFFSET	ATTRIBUTES
IBUFF	000014	INTEGER12 PARAMETER ARRAY (1)
IPUTCH	000016	INTEGER12 VARIABLE
IRELRC	000020	INTEGER12 VARIABLE
ISIL	000062	INTEGER12 VARIABLE
J	000064	INTEGER12 VARIABLE
K	000066	INTEGER12 VARIABLE
L	000068	INTEGER12 PROCEDURE
COMMON BLOCK /FILES/	LENGTH 001134	
ESPEC	000000	INTEGER12 ARRAY (39)
EXT	000116	INTEGER12 ARRAY (2)
NRECS	000122	INTEGER12 VARIABLE
NRECS2	000124	INTEGER12 VARIABLE
ICMIPS	000126	INTEGER12 VARIABLE
IAC	000130	INTEGER12 VARIABLE
NAREC	000132	INTEGER12 VARIABLE
IBUFF	000134	INTEGER12 ARRAY (16,16) VECTORED
INFL	000036	INTEGER12 ARRAY (4)
OUTFL	000036	INTEGER12 ARRAY (4)

PAGE 001

RT-11 FORTRAN IV U018-08A UED 21-MAR-79 17:21:36

0001	0002	FUNCTIONAL DESCRIPTION:
0	0	INTEGER FUNCTION IPRDFM (IBUFF)
0	0	THIS ROUTINE READS ONE FUNCTION MEMORY
0	0	RECORD INTO THE USER BUFFER
0	0	THIS APPLIES ONLY TO THIS ROUTINE:
0	0	INTEGER IBUFF(1)
0	0	NOTE: IBUFF IS DEFINED AS 512 BYTES BY 1
0	0	THIS IS BECAUSE .TRA FILES ARE ONE VALUE
0	0	PER BYTE WHEREAS THE COMTAL FUNCTION MEMORY
0	0	IS ONE VALUE PER WORD (2 BYTES)

```

0003      LOGICALS1 BUFF(512,1)
      . *****
      DATA DEFINITION
      INTEGER INFL(4),OUTFL(4),EXT(2),FSPEC(39)
      EQUIVALENCE (INFL,FSPEC(16)),(OUTFL,FSPEC(16))
      COMMON /IFILES/ FSPEC,EXT,NRECS,NRECSZ,ICHIPS,
      IACT,NABREC,BUFF
      *
      END OF DATA DEFINITION
      *****
      IPRDFM = 0
      NSIZ = 256
      NR = 1
      IC = IREADJ(NSIZ,BUFF(1,1),NR,ICHIPS)
      IF(IC.LT.0) IPRDFM = IC
      ,      CONVERT PER BYTE TO PER WORD
      DO 100 J = 1,256
      IBUFF(J) = BUFF(J,1)
      ,      CORRECT FOR NEGATIVE
      *****

```

```

RT-11 FORTRAN IV      U01B-08A      UED 21-MAR-79 17:21:36      PAGE 002
0015      C      IF(IBUFF(J).LT.0) IBUFF(J) = IBUFF(J) + *400
0017      C      CONTINUE
0018      C      RETURN
0019      C      END

```

STORAGE MAP

OFFSET ATTRIBUTES

```
000014 INTGERR2 PARAMETER  
000016 INTGERR2 VARIABLE  
000022 INTGERR2 VARIABLE  
000024 INTGERR2 VARIABLE  
000026 INTGERR2 VARIABLE  
000030 INTGERR2 PROCEDURE  
000030 INTGERR2 VARIABLE
```

Block /FILES/ LENGTH 001134

```

FPEC      000000    INTEGER12    ARRAY (30)
EXT       000116    INTEGER12    VARIABLE
MRESZ     000122    INTEGER12    VARIABLE
MRESZ     000124    INTEGER12    VARIABLE
IAC1P5    000130    INTEGER12    VARIABLE
IAC1P5     000136    INTEGER12    VARIABLE
MABREC    000132    LOGICAL11    VECTOR
BUFF      000134    INTEGER12    ARRAY (4)
INFL      000036    INTEGER12    ARRAY (4)
OUTFL     000036    INTEGER12    ARRAY (4)

```

PAGE 001

... 2018-09-20 17:21:38

INTERCEPT FUNCTION (INTERCEPT)

FUNCTIONAL DESCRIPTION:

THIS ROUTINE WRITES THE FUNCTION MEMORY RECORD TO THE IPS FILE.

THIS APPLIES ONLY TO THIS ROUTINE:

INTEGER Ibuff(1)

NOTE: BUFF IS DEFINED AS 512 BYTES BY 1
THIS IS BECAUSE .TRA FILES ARE ONE VALUE
PER BYTE WHEREAS THE CORTAL FUNCTION MEMORY
IS ONE VALUE PER WORD (2 BYTES)

```

0003      LOGICAL:1 BUFF(512,1)
      , *****
      DATA DEFINITION
0004      INTEGER INFL(4),OUTFL(4),EXT(2),FSPEC(39)
0005      EQUIVALENCE (INFL,FSPEC(16)),(OUTFL,FSPEC(16))
0006      COMMON /FILES/ FSPEC,EXT,NRECS,NRECSZ,ICHIPS,
      , IACT,NABREC,BUFF
      END OF DATA DEFINITION
      *****
0007      IPUTFM = 0
      , 1) CLEAR BUFFER
0008      DO 100 J=255,512
0009      BUFF(J,1) = 0
0010      100 CONTINUE
      , 2) PACK BLOCK
0011      DO 200 J=1,256
0012      BUFF(J,1) =IBUFF(J)

PT-11 FORTRAN IV      U018-08A      UED 21-MAR-79 17:21:38      PAGE 002

0013      200 CONTINUE
      , 3) WRITE BLOCK TO IPS FILE
      NR=1
0014      NSIZ=256
0015
0016      IC = IPUTFM(NSIZ,BUFF(1,1),NR,ICHIPS)
0017      IF(IC.LT.0) IPUTFM = IC

```

```

C
0010 RETURN
0020 END

```

```

RT-11 FORTRAN IV      STORAGE MAP

NAME  OFFSET  ATTRIBUTES
IBUFF 000014 INTEGER12 PARAMETER ARRAY (1)
IPUTFM 000016 INTEGER12 VARIABLE
J      000022 INTEGER12 VARIABLE
NR      000024 INTEGER12 VARIABLE
NS12    000026 INTEGER12 VARIABLE
IC       000030 INTEGER12 VARIABLE
IURITU  000030 INTEGER12 PROCEDURE

COMMON BLOCK /IFILES/ LENGTH 001134

FSPEC  000000 INTEGER12 ARRAY (39)
EXT     000116 INTEGER12 ARRAY (2)
NRECS   000122 INTEGER12 VARIABLE
NRECSZ  000124 INTEGER12 VARIABLE
ICHIPS  000126 INTEGER12 VARIABLE
IACIT   000130 INTEGER12 VARIABLE
MAREC   000132 INTEGER12 VARIABLE
BUFF    000134 LOGICAL11 ARRAY (512,1) VECTORED
INFL    000036 INTEGER12 ARRAY (4)
OUTFL   000036 INTEGER12 ARRAY (4)

```

FILE NAME: COMLO1.FOR
DATE LAST EDITED: 8-MAR-79

HISTORY:
DATE: 28-FEB-79
8-MAR-79
MODIFICATION:
INITIAL WRITING
CHANGE FUNCTION
NAMES TO START WITH I

CORTAL LIBRARY FUNCTIONS

THIS CORTAL LIBRARY CONSISTS OF A SERIES OF FUNCTION ROUTINES.
THE FUNCTIONS PROVIDE BASIC BUILDING BLOCKS FOR USER PROGRAMS
TO USE THE CORTAL.

THE SOURCE FOR THESE FUNCTIONS ARE IN 5 FILES. THEY ARE:

FILE NAME FUNCTIONS
COMLO1.FOR OPEN AND CLOSE
COMLO2.FOR TRANSFER CODE 1 TYPE
COMLO3.FOR TRANSFER CODE 2 TYPE
COMLO4.FOR TARGET POSITION
COMLO5.FOR TRANSFER CODE 3 TYPE

THE FOLLOWING FUNCTIONS ARE PROVIDED:

NUMBER	NAME	FUNCTION
N/A	CROPEN	START OF CORTAL COMMUNICATION
N/A	CRCLOS	END OF CORTAL COMMUNICATION
1	IUIW	WRITE IMAGE MEMORY
2		NOT DEFINED
3	IRIW	READ IMAGE MEMORY
4	IUCH	WRITE GRAPHIC MEMORY
5		NOT DEFINED
6	IRIW	READ GRAPHIC MEMORY
7	IUPS	WRITE PSEUDO MEMORY

U U U U U U U U

PAGE 002

UUUUUUUUUUUUUUUUUUUUUU

[illegible]

START OF THE LIBRARY ROUTINES

1001

FUNCTIONAL DESCRIPTION:

THIS ROUTINE MUST BE THE FIRST CALL IN A USER PROGRAM FOR CORTAL SERVICE. IT ESTABLISHES

. A CHANNEL - ICHIN - FOR I/O TO THE CONTAL.
 . AFTER THIS ROUTINE IS CALLED, OTHER CONTAL SERVICE
 . ROUTINES MAY BE CALLED. AFTER ALL PROCESSING IS
 . FINISHED, A CALL TO CMCLOS IS MADE TO CLOSE THE CHANNEL.

. ERRORS:
 . 1,2,3 : ERROR IN FETCHING HANDLER
 . -1,-2 : ERROR IN OPENING CONTAL HANDLER

DATA DEFINITION

INTEGER TCUM, TCRIM, TCUGM, TCRGM, TCUPS, TORPS

0002

RT-11 FORTRAN IV U01B-08A FRI 09-MAR-79 16:15:45 PAGE 003

INTEGER TCUM, TCRFM, TCUTP, TCRTP, TC'DC, TORDC
 INTEGER TCUCB, TCRCB, TCUCD, TCRCD, TC'DB, TORDB

0003

C INTEGER INFIL(4),DOB(8,20)

C INTEGER TCC

C INTEGER DB1(8),DB2(8),DB3(8),DB4(8),DB5(8)
 C INTEGER DB6(8),DB7(8),DB8(8),DB9(8),DB10(8)
 C INTEGER DB11(8),DB12(8),DB13(8),DB14(8),DB15(8)
 C INTEGER DB16(8),DB17(8),DB18(8),DB19(8),DB20(8)

LOGICALS1 SPACE

EQUIVALENCE (DB1 , DOB(1,1))
 EQUIVALENCE (DB2 , DOB(1,2))
 EQUIVALENCE (DB3 , DOB(1,3))
 EQUIVALENCE (DB4 , DOB(1,4))
 EQUIVALENCE (DB5 , DOB(1,5))
 EQUIVALENCE (DB6 , DOB(1,6))
 EQUIVALENCE (DB7 , DOB(1,7))
 EQUIVALENCE (DB8 , DOB(1,8))
 EQUIVALENCE (DB9 , DOB(1,9))
 EQUIVALENCE (DB10, DOB(1,10))
 EQUIVALENCE (DB11, DOB(1,11))
 EQUIVALENCE (DB12, DOB(1,12))
 EQUIVALENCE (DB13, DOB(1,13))

0012
 0013
 0014
 0015
 0016
 0017
 0018
 0019
 0020
 0021
 0022
 0023
 0024


```

      SET FUNCTION = SUCCESS
      ICMOPE = 0
      0069
      000000
      1)
      ICHIN=IGETC(
      0070
      2)
      IC=IFETCH(INFILE)
      IF(IC.NE.0) GO TO 100
      0071
      0072
      3)
      IC=LOOKUP(ICHIN,INFILE)
      0074
      IF(IC.LT.0) ICMOPE=IC
      0075
      RETURN
      0077
      END
      0078

```

RT-11 FORTRAN IV STORAGE MAP

NAME	OFFSET	ATTRIBUTES
INFILE	000014	INTEGER12 ARRAY (4)
ICMOPE	000024	INTEGER12 VARIABLE
SPACE	000026	LOGICAL11 VARIABLE
IGETC	000000	INTEGER12 PROCEDURE
IC	000100	INTEGER12 VARIABLE
IFETCH	000000	INTEGER12 PROCEDURE
LOOKUP	000000	INTEGER12 PROCEDURE
COMMON BLOCK /DDB/ LENGTH 001160		
DDB	000000	INTEGER12 ARRAY (8,20) VECTORED
DB1	000000	INTEGER12 ARRAY (8)
DB2	000020	INTEGER12 ARRAY (8)
DB3	000040	INTEGER12 ARRAY (8)
DB4	000060	INTEGER12 ARRAY (8)
DB5	000100	INTEGER12 ARRAY (8)
DB6	000120	INTEGER12 ARRAY (8)
DB7	000140	INTEGER12 ARRAY (8)
DB8	000160	INTEGER12 ARRAY (8)
DB9	000200	INTEGER12 ARRAY (8)
DB10	000220	INTEGER12 ARRAY (8)
DB11	000240	INTEGER12 ARRAY (8)
DB12	000260	INTEGER12 ARRAY (8)

```

DE13 000300 INTEGER12 ARRAY (8)
DE14 000320 INTEGER12 ARRAY (8)
DE15 000340 INTEGER12 ARRAY (8)
DE16 000360 INTEGER12 ARRAY (8)
DE17 000400 INTEGER12 ARRAY (8)
DE18 000420 INTEGER12 ARRAY (8)
DE19 000440 INTEGER12 ARRAY (8)
DE20 000460 INTEGER12 ARRAY (8)
COMMON BLOCK /TC/ LENGTH 000044

```

```

TCUIM 000000 INTEGER12 VARIABLE
TCRIM 000002 INTEGER12 VARIABLE
TCUGM 000004 INTEGER12 VARIABLE
TCRGM 000006 INTEGER12 VARIABLE
TCUPS 000010 INTEGER12 VARIABLE
TCRPS 000012 INTEGER12 VARIABLE
TCUFH 000014 INTEGER12 VARIABLE
TCRFH 000016 INTEGER12 VARIABLE
TCUTP 000020 INTEGER12 VARIABLE
TCRTP 000022 INTEGER12 VARIABLE
TCUDC 000024 INTEGER12 VARIABLE
TCRDC 000026 INTEGER12 VARIABLE
TCUCB 000030 INTEGER12 VARIABLE
TCRCB 000032 INTEGER12 VARIABLE
TCUCD 000034 INTEGER12 VARIABLE
TCRCD 000036 INTEGER12 VARIABLE
TCUDB 000040 INTEGER12 VARIABLE
TCRDB 000042 INTEGER12 VARIABLE
COMMON BLOCK /CHAN/ LENGTH 000002

```

```

ICHIN 000000 INTEGER12 VARIABLE

```

```

RT-11 FORTRAN IV STORAGE MAP
NAME OFFSET ATTRIBUTES

```

```

COMMON BLOCK /COMVAR/ LENGTH 000006
MAXIM 000000 INTEGER12 VARIABLE
MAXLN 000002 INTEGER12 VARIABLE
MAXGM 000004 INTEGER12 VARIABLE
COMMON BLOCK /DATAA/ LENGTH 000002
TCC 000000 INTEGER12 VARIABLE

```

```

0001 C      INTEGER FUNCTION ICMCLO
      C      FUNCTIONAL DESCRIPTION:
      C      THIS ROUTINE IS CALLED AFTER ALL CORTAL
      C      PROCESSING IS FINISHED. A CALL TO ICMCLO MUST
      C      HAVE BEEN CALLED PRIOR TO THIS CALL.
      C      ERRORS:
      C      NONE.
      C      *****
      C      DATA DEFINITION
      C      INTEGER TCUIR, TCRIH, TCUGH, TCRGH, TCUPS, TCPS
      C      INTEGER TCUPH, TCRPH, TCUTP, TCRTP, TCUDC, TCRDC
      C      INTEGER TCUCB, TCRCB, TCUCD, TCRCD, TCUBD, TCRBD
      C      INTEGER DBB(8,20)
      C      INTEGER TCC
      C      COMMON /DBB/ DBB
      C      COMMON /TC/ TCUIR, TCRIH, TCUGH, TCRGH, TCUPS, TCPS
      C      TCUPH, TCRPH, TCUTP, TCRTP, TCUDC, TCRDC
      C      TCUCB, TCRCB, TCUCD, TCRCD, TCUBD, TCRBD
      C      COMMON /CHAN/ ICHIN
      C      COMMON /COMVAR/ MAXIR, MAXIH, MAXGH
      C      COMMON /DATAA/ TCC
      C      END OF DATA DEFINITION
      C      NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION ICMCLO
      C      -----
      C      *****
      C      ICMCLO=0
  
```

```

0013      CALL CLOSEC(ICHIN)
0014      RETURN
0015      END

```

```

RT-11 FORTRAN IV      STORAGE MAP

NAME      OFFSET      ATTRIBUTES

ICMCL0    000014      INTEGER12 VARIABLE
CLOSEC    000000      REAL14  PROCEDURE

COMMON BLOCK /DDB/      LENGTH 000500

DDB      000000      INTEGER12 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/      LENGTH 000044

TCUIM      000000      INTEGER12 VARIABLE
TCRIM      000002      INTEGER12 VARIABLE
TCUGM      000004      INTEGER12 VARIABLE
TCRGM      000006      INTEGER12 VARIABLE
TCUPS      000010      INTEGER12 VARIABLE
TCRPS      000012      INTEGER12 VARIABLE
TCUFM      000014      INTEGER12 VARIABLE
TCRFM      000016      INTEGER12 VARIABLE
TCUTP      000020      INTEGER12 VARIABLE
TCRTP      000022      INTEGER12 VARIABLE
TCUDC      000024      INTEGER12 VARIABLE
TCRDC      000026      INTEGER12 VARIABLE
TCUCB      000030      INTEGER12 VARIABLE
TCRCB      000032      INTEGER12 VARIABLE
TCUCD      000034      INTEGER12 VARIABLE
TCPCD      000036      INTEGER12 VARIABLE
TCUDB      000040      INTEGER12 VARIABLE
TCRDB      000042      INTEGER12 VARIABLE

COMMON BLOCK /CHAN/      LENGTH 000002

ICHIN      000000      INTEGER12 VARIABLE

COMMON BLOCK /COMVAR/      LENGTH 000006

MAXIM      000000      INTEGER12 VARIABLE
MAXLN      000002      INTEGER12 VARIABLE
MAXGM      000004      INTEGER12 VARIABLE

COMMON BLOCK /DATAA/      LENGTH 000002

TCC      000000      INTEGER12 VARIABLE

```

RT-11 FORTRAN IV U01B-08A FRI 09-MAR-79 16:16:01 PAGE 001

FILE NAME: COM102.FOR
DATE LAST EDITED: 8-MAR-79

HISTORY:
DATE: MODIFICATION

28-FEB-79 INITIAL WRITING
8-MAR-79 CHANGE FUNCTION
 NAMES TO START WITH I

INTEGER FUNCTION IJIN(IJNO, IJLN, BUFF)

FUNCTIONAL DESCRIPTION:

THIS FUNCTION IS USED TO WRITE ONE LINE OF
DATA FROM USER MEMORY TO THE COMITAL IMAGE SELECTED.

PARAMETERS:

IJNO : THE IMAGE NUMBER (1-3)
IJLN : THE LINE NUMBER (1-256) 1 AT TOP, 256 AT BOTTOM
BUFF : USER BUFFER ADDRESS

ERRORS:

0 : SUCCESS
-1 : WRITE ERROR

DATA DEFINITION

LOGICAL*1 BUFF(256)

INTEGER TCUM, TCUMN, TCUMH, TCUMH, TCUMS, TCUMS
INTEGER TCUMN, TCUMN, TCUMN, TCUTP, TCUDC, TCUDC
INTEGER TCUCB, TCUCB, TCUCB, TCUCB, TCUCB, TCUCB

0002

0003
0004
0005

NAME	OFFSET	ATTRIBUTES
BUFF	000020	LOGICALS11 PARAMETER ARRAY (256)
ICUIN	000022	INTEGER32 VARIABLE
IMAC	000014	INTEGER32 PARAMETER VARIABLE
ITC1	000016	INTEGER32 PARAMETER VARIABLE
ITC1	000000	INTEGER32 PROCEDURE
COMMON BLOCK /DDB/ LENGTH 000500		
DDB	000000	INTEGER32 ARRAY (8,20) VECTORED
COMMON BLOCK /TC/ LENGTH 000044		
TCUIN	000000	INTEGER32 VARIABLE
TCUIN	000002	INTEGER32 VARIABLE
TCUIN	000004	INTEGER32 VARIABLE
TCUGA	000006	INTEGER32 VARIABLE
TCUGA	000010	INTEGER32 VARIABLE
TCUPS	000012	INTEGER32 VARIABLE
TCUPS	000014	INTEGER32 VARIABLE
TCUFN	000016	INTEGER32 VARIABLE
TCUFN	000020	INTEGER32 VARIABLE
TCUTP	000022	INTEGER32 VARIABLE
TCUTP	000024	INTEGER32 VARIABLE
TCUDC	000026	INTEGER32 VARIABLE
TCUDC	000030	INTEGER32 VARIABLE
TCUCB	000032	INTEGER32 VARIABLE
TCUCB	000034	INTEGER32 VARIABLE
TCUCD	000036	INTEGER32 VARIABLE
TCUCD	000040	INTEGER32 VARIABLE
TCURB	000042	INTEGER32 VARIABLE
COMMON BLOCK /CHAW/ LENGTH 000002		
ICHIN	000000	INTEGER32 VARIABLE
COMMON BLOCK /COMVAR/ LENGTH 000006		
MAXIN	000000	INTEGER32 VARIABLE
MAXIN	000002	INTEGER32 VARIABLE
MAXGA	000004	INTEGER32 VARIABLE
COMMON BLOCK /DATAA/ LENGTH 000002		
TCC	000000	INTEGER32 VARIABLE

SET FUNCTION CODE

TCC-TCRIM

CALL TRANSFER CODE 1 COMMON ROUTINE

IRIM-ITCI (IMNO, IMLN, BUFF)

0015
0016

PT-11 FORTRAN IV STORAGE MAP

NAME	OFFSET	ATTRIBUTES
...

```

000020 LOGICAL11 PARAMETER ARRAY (256)
000022 INTEGER22 VARIABLE
000014 INNO INTEGER32 PARAMETER VARIABLE
000016 IRLN INTEGER32 PARAMETER VARIABLE
000000 ITC1 INTEGER32 PROCEDURE

```

COMMON BLOCK /DDB/ LENGTH 000500

```

DDE 000000 INTEGER12 ARRAY (8,20) VECTORED

```

COMMON BLOCK /TC/	LENGTH 000044
000000	000000
000001	000001
000002	000002
000003	000003
000004	000004
000005	000005
000006	000006
000007	000007
000008	000008
000009	000009
000010	000010
000011	000011
000012	000012
000013	000013
000014	000014
000015	000015
000016	000016
000017	000017
000018	000018
000019	000019
000020	000020
000021	000021
000022	000022
000023	000023
000024	000024
000025	000025
000026	000026
000027	000027
000028	000028
000029	000029
000030	000030
000031	000031
000032	000032
000033	000033
000034	000034
000035	000035
000036	000036
000037	000037
000038	000038
000039	000039
000040	000040
000041	000041
000042	000042
000043	000043
000044	000044
000045	000045
000046	000046
000047	000047
000048	000048
000049	000049
000050	000050
000051	000051
000052	000052
000053	000053
000054	000054
000055	000055
000056	000056
000057	000057
000058	000058
000059	000059
000060	000060
000061	000061
000062	000062
000063	000063
000064	000064
000065	000065
000066	000066
000067	000067
000068	000068
000069	000069
000070	000070
000071	000071
000072	000072
000073	000073
000074	000074
000075	000075
000076	000076
000077	000077
000078	000078
000079	000079
000080	000080
000081	000081
000082	000082
000083	000083
000084	000084
000085	000085
000086	000086
000087	000087
000088	000088
000089	000089
000090	000090
000091	000091
000092	000092
000093	000093
000094	000094
000095	000095
000096	000096
000097	000097
000098	000098
000099	000099
000100	000100
000101	000101
000102	000102
000103	000103
000104	000104
000105	000105
000106	000106
000107	000107
000108	000108
000109	000109
000110	000110
000111	000111
000112	000112
000113	000113
000114	000114
000115	000115
000116	000116
000117	000117
000118	000118
000119	000119
000120	000120
000121	000121
000122	000122
000123	000123
000124	000124
000125	000125
000126	000126
000127	000127
000128	000128
000129	000129
000130	0

000000	INTEGER2	VARIABLE
000001	INTEGER2	VARIABLE
000002	INTEGER2	VARIABLE
000003	INTEGER2	VARIABLE
000004	INTEGER2	VARIABLE
000005	INTEGER2	VARIABLE
000006	INTEGER2	VARIABLE
000007	INTEGER2	VARIABLE
000008	INTEGER2	VARIABLE
000009	INTEGER2	VARIABLE
000010	INTEGER2	VARIABLE
000011	INTEGER2	VARIABLE
000012	INTEGER2	VARIABLE
000013	INTEGER2	VARIABLE
000014	INTEGER2	VARIABLE
000015	INTEGER2	VARIABLE
000016	INTEGER2	VARIABLE
000017	INTEGER2	VARIABLE
000018	INTEGER2	VARIABLE
000019	INTEGER2	VARIABLE
000020	INTEGER2	VARIABLE

```

TCNTP 000022 INTEGER12 VARIABLE
TCN1 000024 INTEGER12 VARIABLE
TCN2 000026 INTEGER12 VARIABLE
TCN3 000028 INTEGER12 VARIABLE
TCN4 000030 INTEGER12 VARIABLE
TCN5 000032 INTEGER12 VARIABLE
TCN6 000034 INTEGER12 VARIABLE
TCN7 000036 INTEGER12 VARIABLE
TCN8 000038 INTEGER12 VARIABLE
TCN9 000040 INTEGER12 VARIABLE
TCN10 000042 INTEGER12 VARIABLE
COMMON BLOCK /CHNN/ LENGTH 000002
ICN1 000000 INTEGER12 VARIABLE
COMMON BLOCK /CONNR/ LENGTH 000006
MAX1 000000 INTEGER12 VARIABLE
MAX2 000002 INTEGER12 VARIABLE
MAX3 000004 INTEGER12 VARIABLE
COMMON BLOCK /DATA/ LENGTH 000002
TCC 000000 INTEGER12 VARIABLE

```

RT-11 FORTRAN IV U018-08A FRI 09-MAR-79 16:16:05 PAGE 001

```

0001 C INTEGER FUNCTION IUCN(IMNO,IMLN,BUFF)
      C
      C FUNCTIONAL DESCRIPTION:
      C
      C THIS FUNCTION IS USED TO WRITE ONE LINE OF
      C DATA FROM USER MEMORY TO THE CORTAL GRAPHIC SELECTED.
      C
      C PARAMETERS:
      C
      C IMNO : THE GRAPHIC NUMBER (1-3)
      C IMLN : THE LINE NUMBER (1-256) 1 AT TOP, 256 AT BOTTOM
      C BUFF : USER BUFFER ADDRESS
      C
      C ERRORS:
      C
      C 0 : SUCCESS
      C -1 : WRITE ERROR

```

RT-11 FORTRAN IV		STORAGE MAP	
NAME	OFFSET	ATTRIBUTES	
BUFF	000020	LOGICALS11	PARAMETER ARRAY (256)
IUCH	000022	INTEGER12	VARIABLE
IMPO	000014	INTEGER12	PARAMETER VARIABLE
IFLN	000016	INTEGER12	PARAMETER VARIABLE
ITC1	000000	INTEGER12	PROCEDURE
COMMON BLOCK /DDB/ LENGTH 000500			
DDB	000000	INTEGER12	ARRAY (8,20) VECTORED
COMMON BLOCK /TC/ LENGTH 000044			
TCUIM	000000	INTEGER12	VARIABLE
TCUIM	000002	INTEGER12	VARIABLE
TCUIM	000004	INTEGER12	VARIABLE
TCUIM	000006	INTEGER12	VARIABLE
TCUPS	000010	INTEGER12	VARIABLE
TCUPS	000012	INTEGER12	VARIABLE
TCUFA	000014	INTEGER12	VARIABLE
TCUFA	000016	INTEGER12	VARIABLE
TCUFP	000020	INTEGER12	VARIABLE
TCUFP	000022	INTEGER12	VARIABLE
TCUJC	000024	INTEGER12	VARIABLE
TCUJC	000026	INTEGER12	VARIABLE
TCUCB	000030	INTEGER12	VARIABLE
TCUCB	000032	INTEGER12	VARIABLE
TCUCD	000034	INTEGER12	VARIABLE
TCUCD	000036	INTEGER12	VARIABLE
TCUCB	000040	INTEGER12	VARIABLE
TCUCB	000042	INTEGER12	VARIABLE
COMMON BLOCK /CHAN/ LENGTH 000002			
ICHIN	000000	INTEGER12	VARIABLE
COMMON BLOCK /COMVAR/ LENGTH 000006			
MMVIN	000000	INTEGER12	VARIABLE
MMVIN	000002	INTEGER12	VARIABLE
MMVIN	000004	INTEGER12	VARIABLE
COMMON BLOCK /DATA00/ LENGTH 000002			
TCC	000000	INTEGER12	VARIABLE


```

0011 C COMMON /COMMON/ MAXIM,MAXLN,MAXGM
0012 C COMMON /DATA/ TCC
0013 C END OF DATA DEFINITION
0014 C
NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CROPER

```

RT-11 FORTRAN IV U01B-08A FRI 09-MAR-79 16:16:07 PAGE 002

```

0013 C *****
0014 C . SET FUNCTION CODE
0015 C TCC=TCRGM
0016 C . CALL TRANSFER CODE 1 COMMON ROUTINE
0017 C IRGM=ITC1(IMMO,IMLN,BUFF)
0018 C RETURN
0019 C END

```

```

RT-11 FORTRAN IV STORAGE MAP
NAME OFFSET ATTRIBUTES
BUFF 000020 LOGICAL*1 PARAMETER ARRAY (256)
IRGM 000022 INTEGER*2 VARIABLE
IMMO 000014 INTEGER*2 PARAMETER VARIABLE
IMLN 000016 INTEGER*2 PARAMETER VARIABLE
ITC1 000000 INTEGER*2 PROCEDURE
COMMON BLOCK /DDB/ LENGTH 000500
DDB 000000 INTEGER*2 ARRAY (8,20) VECTORED
COMMON BLOCK /TC/ LENGTH 000044

```


NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CHOPEN

RT-11 FORTRAN IV STORAGE MAP

NAME OFFSET ATTRIBUTES

000020 LOGICAL11 PARAMETER ARRAY (256)
 000022 INTEGER12 VARIABLE
 000014 INTEGER12 PARAMETER VARIABLE
 000016 INTEGER12 PARAMETER VARIABLE
 000100 INTEGER12 VARIABLE
 000102 INTEGER12 VARIABLE

```

IADOP 000000 INTEGER12 PROCEDURE
IC 000104 INTEGER12 VARIABLE
IUPITU 000100 INTEGER12 PROCEDURE

COMMON BLOCK /DDB/ LENGTH 000500

DDS 000000 INTEGER12 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/ LENGTH 000044
TCUIN 000000 INTEGER12 VARIABLE
TCRIN 000002 INTEGER12 VARIABLE
TCUIN 000004 INTEGER12 VARIABLE
TCRIN 000006 INTEGER12 VARIABLE
TCUPS 000010 INTEGER12 VARIABLE
TCRPS 000012 INTEGER12 VARIABLE
TCUIN 000014 INTEGER12 VARIABLE
TCRIN 000016 INTEGER12 VARIABLE
TCUTP 000020 INTEGER12 VARIABLE
TCRTP 000022 INTEGER12 VARIABLE
TCUDC 000024 INTEGER12 VARIABLE
TCRDC 000026 INTEGER12 VARIABLE
TCUCB 000030 INTEGER12 VARIABLE
TCRCB 000032 INTEGER12 VARIABLE
TCUCD 000034 INTEGER12 VARIABLE
TCRCD 000036 INTEGER12 VARIABLE
TCUDB 000040 INTEGER12 VARIABLE
TCRDB 000042 INTEGER12 VARIABLE

COMMON BLOCK /CHAN/ LENGTH 000002
ICHIN 000000 INTEGER12 VARIABLE

COMMON BLOCK /COMVAR/ LENGTH 000006
MAXIN 000000 INTEGER12 VARIABLE
MAXLN 000002 INTEGER12 VARIABLE
MAXGM 000004 INTEGER12 VARIABLE

COMMON BLOCK /DATAa/ LENGTH 000002
TCC 000000 INTEGER12 VARIABLE

```

RT-11 FORTRAN IV U018-08A UED 21-MAR-79 16:28:39

FILE NAME: COM03.FOR
 DATE LAST EDITED: 21-MAR-79
 HISTORY:
 DATE: 28-FEB-79
 19-MAR-79
 21-MAR-79
 MODIFICATION

 INITIAL WRITING
 CHANGE FUNCTIONS
 TO START WITH 1
 CHANGE ALL 'FMNO'
 TO 'IFMNO'

INTEGER FUNCTION IUFM(IFMNO,BUFF)

FUNCTIONAL DESCRIPTION:

THIS ROUTINE WRITES THE SELECTED FUNCTION
 MEMORY FROM THE USER BUFFER.

PARAMETERS:

IFMNO - FUNCTION MEMORY NUMBER (1-3)
 BUFF - USER BUFFER (INTEGER ARRAY 256)

ERRORS:

0 - SUCCESS
 -1 - WRITE ERROR

DATA DEFINITION

INTEGER BUFF(256)

INTEGER TCUIA, TCRIH, TCRH, TCRH, TCUPS, TCPS
 INTEGER TCUPH, TCRFH, TCRTP, TCRTP, TCRDC, TCRDC
 INTEGER TCUCB, TCRCB, TCUCD, TCRCD, TCUCB, TCRDB

INTEGER DOB(8,20)

0001

0001

0002

0003
 0004
 0005

0006

PAGE 002

WED 21-MAR-79 16:28:39

U01B-08A

RT-11 FORTRAN IV

```

1) SET FUNCTION NUMBER
2) CALL TRANSFER CODE 2 COMMON ROUTINE

```

TCC-TCUFM

IUFM-ITC2(IFMNO, BUFF)

RETURN

END

PT-1: FORTRAN IV STORAGE MAP

NAME OFFSET ATTRIBUTES

```

000016 INTEGER12 PARAMETER ARRAY (256)
000020 INTEGER12 VARIABLE
000014 INTEGER12 PARAMETER VARIABLE

```

```

ITC2 000000 INTEGER12 PROCEDURE
COMMON BLOCK /DDB/      LENGTH 000500

DDB 000000 INTEGER12 ARRAY (8,20) VECTORED
COMMON BLOCK /TC/      LENGTH 000044
TCUIN 000000 INTEGER12 VARIABLE
TCUIN 000002 INTEGER12 VARIABLE
TCUIN 000004 INTEGER12 VARIABLE
TCUIN 000006 INTEGER12 VARIABLE
TCUIN 000010 INTEGER12 VARIABLE
TCUIN 000012 INTEGER12 VARIABLE
TCUIN 000014 INTEGER12 VARIABLE
TCUIN 000016 INTEGER12 VARIABLE
TCUIN 000020 INTEGER12 VARIABLE
TCUIN 000022 INTEGER12 VARIABLE
TCUIN 000024 INTEGER12 VARIABLE
TCUIN 000026 INTEGER12 VARIABLE
TCUIN 000030 INTEGER12 VARIABLE
TCUIN 000032 INTEGER12 VARIABLE
TCUIN 000034 INTEGER12 VARIABLE
TCUIN 000036 INTEGER12 VARIABLE
TCUIN 000040 INTEGER12 VARIABLE
TCUIN 000042 INTEGER12 VARIABLE
COMMON BLOCK /CHAN/      LENGTH 000002
TCUIN 000000 INTEGER12 VARIABLE
COMMON BLOCK /COMVAR/      LENGTH 000006
MAXIN 000000 INTEGER12 VARIABLE
MAXIN 000002 INTEGER12 VARIABLE
MAXIN 000004 INTEGER12 VARIABLE
COMMON BLOCK /DATA0/      LENGTH 000002
TCC 000000 INTEGER12 VARIABLE

```

PAGE 001

PT-11 FORTPMN IU U01B-08A WED 21-MAR-79 16:28:41

```

C
0001 C INTEGER FUNCTION IRFH(IFNO,BUFF)
C
C FUNCTIONAL DESCRIPTION:
C
C THIS ROUTINE READS THE SELECTED FUNCTION
C MEMORY AND FILLS THE USER BUFFER.
C

```



```

C      1) SET FUNCTION NUMBER
C      2) CALL TRANSFER CODE 2 COMMON ROUTINE
C
C      TCC=TCRFH
C      IRFM=ITC2(IFMNO,BUFF)
C      RETURN
C      END
0013
0014
0015
0016

```

RT-11 FORTRAN IV STORAGE MAP

NAME OFFSET ATTRIBUTES

```

BUFF 000016 INTEGER12 PARAMETER ARRAY (256)
IRFM 000020 INTEGER12 VARIABLE
IFMNO 000014 INTEGER12 PARAMETER VARIABLE
ITC2 000000 INTEGER12 PROCEDURE

```

COMMON BLOCK /DDB/ LENGTH 000500

DDB 000000 INTEGER12 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/ LENGTH 000044

```

TCUIM 000000 INTEGER12 VARIABLE
TCPIH 000002 INTEGER12 VARIABLE
TCUIM 000004 INTEGER12 VARIABLE
TCUIM 000006 INTEGER12 VARIABLE
TCUIM 000010 INTEGER12 VARIABLE
TCUIM 000012 INTEGER12 VARIABLE
TCUIM 000014 INTEGER12 VARIABLE
TCUIM 000016 INTEGER12 VARIABLE
TCUIM 000020 INTEGER12 VARIABLE
TCUIM 000022 INTEGER12 VARIABLE
TCUIM 000024 INTEGER12 VARIABLE
TCUIM 000026 INTEGER12 VARIABLE
TCUIM 000030 INTEGER12 VARIABLE
TCUIM 000032 INTEGER12 VARIABLE
TCUIM 000034 INTEGER12 VARIABLE
TCUIM 000036 INTEGER12 VARIABLE
TCUIM 000040 INTEGER12 VARIABLE
TCUIM 000042 INTEGER12 VARIABLE

```

COMMON BLOCK /CHAN/ LENGTH 000002

ICMIM 000000 INTEGER12 VARIABLE

COMMON BLOCK /COMVAR/ LENGTH 000006

```

000000 INTEGER12 VARIABLE
P14 LN 000002 INTEGER12 VARIABLE
P14 LN 000004 INTEGER12 VARIABLE
COMMON BLOCK /DATA/ LENGTH 000002
TCC 000000 INTEGER12 VARIABLE

```

PT-11 FORTRAN IV U01B-08A UED 21-MAR-79 16:28:44 PAGE 001

```

C
0001 C INTEGER FUNCTION IUPS(BUFF)
C
C FUNCTIONAL DESCRIPTION:
C THIS ROUTINE WRITES THE PSEUDO-COLOR
C MEMORY FROM THE USER BUFFER.
C
C PARAMETERS:
C BUFF - USER BUFFER (INTEGER ARRAY 256)
C
C ERRORS:
C 0 - SUCCESS
C -1 - WRITE ERROR
C
C *****
C DATA DEFINITION
C
C002 C INTEGER BUFF(256)
C
C003 C INTEGER TCUIN, TCRIN, TCUH, TCRH, TCRGM, TCRGM, TCRPS, TCRPS
C004 C INTEGER TCUFM, TCRFM, TCUFP, TCRFP, TCUUC, TCRUC, TCRDC, TCRDC
C005 C INTEGER TCUUB, TCRUB, TCUUD, TCRUD, TCUDB, TCRDB
C
C006 C INTEGER DDB(8,20)
C007 C INTEGER TCC
C
C008 C COMMON /DDB/ DDB
C009 C COMMON /TC/ TCUIN, TCRIN, TCUH, TCRH, TCRGM, TCRGM, TCRPS, TCRPS,
C TCUFM, TCRFM, TCUFP, TCRFP, TCUUC, TCRUC, TCRDC, TCRDC,
C TCUUB, TCRUB, TCUUD, TCRUD, TCUDB, TCRDB

```

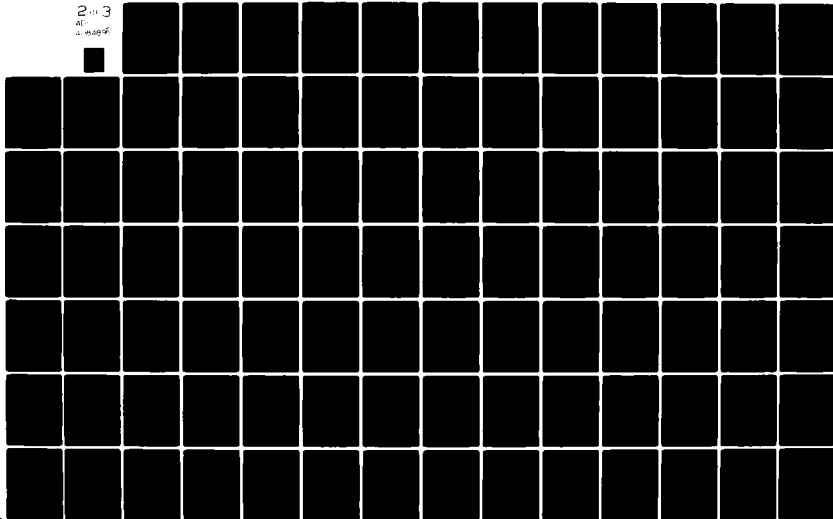
AD-A084 895

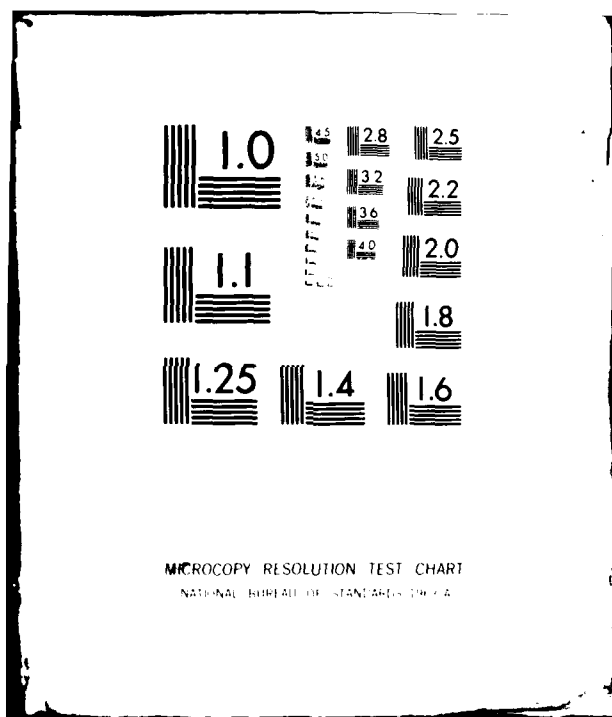
MEASUREMENT CONCEPT CORP ROME NY F/G 17/9
TACTICAL TARGET IDENTIFICATION (TTI) LABORATORY SIMULATION. VOL--ETC(U)
MAR 80 R KELLOGG, R A MCKIE, J TIMS F30602-79-C-0029
RADC -TR-80-41-VOL-1 NL

UNCLASSIFIED

2-113

ALC
A-00000





```

C      COMMON /CHAM/ ICHIN
C      COMMON /CONVAR/ MAXIN,MAXLN,MAXGM
C      COMMON /DATAA/ TCC
C      END OF DATA DEFINITION
C      NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CHOPEN
C      -----
C      *****
C      1) SET FUNCTION NUMBER

```

PAGE 002

```

RT-11 FORTRAN IV      U01B-08A      WED 21-MAR-79 16:28:44
C      2) CALL TRANSFER CODE 2 COMMON ROUTINE
C      TCC-TCUPS
C      IUPS-ITC2(IFMNO,BUFF)
C      RETURN
C      END

```

```

RT-11 FORTRAN IV      STORAGE MAP
NAME  OFFSET  ATTRIBUTES
BUFF  000014  INTEGER12 PARAMETER ARRAY (256)
IUPS  000016  INTEGER12 VARIABLE
ITC2  000000  INTEGER12 PROCEDURE
IFMNO 000070  INTEGER12 VARIABLE
COMMON BLOCK /DDB/      LENGTH 000500
DDB  000000  INTEGER12 ARRAY (8,20) VECTORED
COMMON BLOCK /TC/      LENGTH 000044
TCUIN 000000  INTEGER12 VARIABLE

```

```

TCUIN 000002 INTEGER12 VARIABLE
TCUIN 000004 INTEGER12 VARIABLE
TCUIN 000006 INTEGER12 VARIABLE
TCUIN 000010 INTEGER12 VARIABLE
TCUIN 000012 INTEGER12 VARIABLE
TCUIN 000014 INTEGER12 VARIABLE
TCUIN 000016 INTEGER12 VARIABLE
TCUIN 000020 INTEGER12 VARIABLE
TCUIN 000022 INTEGER12 VARIABLE
TCUIN 000024 INTEGER12 VARIABLE
TCUIN 000026 INTEGER12 VARIABLE
TCUIN 000030 INTEGER12 VARIABLE
TCUIN 000032 INTEGER12 VARIABLE
TCUIN 000034 INTEGER12 VARIABLE
TCUIN 000036 INTEGER12 VARIABLE
TCUIN 000040 INTEGER12 VARIABLE
TCUIN 000042 INTEGER12 VARIABLE
COMMON BLOCK /CHAN/ LENGTH 000002
ICHIN 000000 INTEGER12 VARIABLE
COMMON BLOCK /COMVAR/ LENGTH 000006
MAXIN 000000 INTEGER12 VARIABLE
MAXIN 000002 INTEGER12 VARIABLE
MAXIN 000004 INTEGER12 VARIABLE
COMMON BLOCK /DATA/ LENGTH 000002
TCC 000000 INTEGER12 VARIABLE

```

PAGE 001

PT-11 FORTRAN IU U01B-08A UED 21-MAR-79 16:28:47

```

0001 0 INTEGER FUNCTION IRPS(BUFF)
      0 FUNCTIONAL DESCRIPTION:
      0 THIS ROUTINE READS THE PSEUDO-COLOR
      0 MEMORY AND PUTS IT IN THE USER BUFFER.
      0 PARAMETERS:
      0 BUFF = USER BUFFER (INTEGER ARRAY 256)
      0 ERRORS:
      0 0 = SUCCESS
      0 -1 = WRITE ERROR

```


RT-11	FORTAN IV	STORAGE MAP
NAME	OFFSET	ATTRIBUTES
BLFF	000014	INTEGER12 PARAMETER ARRAY (256)
IRPS	000016	INTEGER12 VARIABLE
ITC2	000000	INTEGER12 PROCEDURE
IFMNO	000070	INTEGER12 VARIABLE
COMMON BLOCK /DBB/ LENGTH 000500		
DBB	000000	INTEGER12 ARRAY (8,20) VECTORED
COMMON BLOCK /TC/ LENGTH 000044		
TCUIM	000000	INTEGER12 VARIABLE
TCRIM	000002	INTEGER12 VARIABLE
TCUGM	000004	INTEGER12 VARIABLE
TCRGM	000006	INTEGER12 VARIABLE
TCUPS	000010	INTEGER12 VARIABLE
TCRPS	000012	INTEGER12 VARIABLE
TCUFM	000014	INTEGER12 VARIABLE
TCRFM	000016	INTEGER12 VARIABLE
TCUTP	000020	INTEGER12 VARIABLE
TCRTP	000022	INTEGER12 VARIABLE
TCUDC	000024	INTEGER12 VARIABLE
TCRDC	000026	INTEGER12 VARIABLE
TCUCB	000030	INTEGER12 VARIABLE
TCRCB	000032	INTEGER12 VARIABLE
TCUCD	000034	INTEGER12 VARIABLE
TCRCD	000036	INTEGER12 VARIABLE
TCUDB	000040	INTEGER12 VARIABLE
TCRDB	000042	INTEGER12 VARIABLE
COMMON BLOCK /CMAN/ LENGTH 000002		
ICMIM	000000	INTEGER12 VARIABLE
COMMON BLOCK /COMVAR/ LENGTH 000006		
MAXIM	000000	INTEGER12 VARIABLE
MAXIM	000002	INTEGER12 VARIABLE
MAXCM	000004	INTEGER12 VARIABLE
COMMON BLOCK /DATAA/ LENGTH 000002		
TCC	000000	INTEGER12 VARIABLE

0001 C INTEGER FUNCTION ITC2(IFMNO,BUFF)

C FUNCTIONAL DESCRIPTION:

C THIS ROUTINE IS CALLED BY FUNCTION MEMORY , AND PSEUDO
C COLOR MEMORY, READ/WRITE

C CALLER SETS THE FUNCTION NUMBER

C THE ROUTINE:

- 1) DETERMINES WHETHER PSEUDO OR FUNCTION (ONLY 1 PSEUDO MEMORY)
- 2) FORCES FUNCTION MEMORY < 4
- 3) SETS COMMAND FOR TRANSFER CODE 2
- 4) CALLS HANDLER

C PARAMETERS:

C IFMNO = FUNCTION MEMORY NUMBER (1-3)

C BUFF = USER BUFFER (INTEGER ARRAY 256)

C ERRORS:

- 0 = SUCCESS
- 1 = WRITE ERROR

C DATA DEFINITION

C INTEGER BUFF(256)

C INTEGER TCUIH, TCRIN, TCUH, TCRH, TCRN, TCUPS, TCPS,
C INTEGER TCUFH, TCRFH, TCUTP, TCRTP, TCUDC, TCDBC,
C INTEGER TCUCB, TCRCB, TCUCD, TCRCD, TCUDB, TCDBB

C INTEGER DDB(8,20)
C INTEGER TCC

C COMMON /DDB/ DDB

C COMMON /TC/ TCUIH, TCRIN, TCUH, TCRH, TCRN, TCUPS, TCPS,
C TCUFH, TCRFH, TCUTP, TCRTP, TCUDC, TCDBC, TCRDC,
C

```

      TCUCB, TCRCB, TCUCD, TCACD, TCUCB, TCRCB
      COMMON /CHAN/ ICHIN
      COMMON /CONVAR/ MAXIN, MAXLN, MAXGM

```

RT-11 FORTRAN IV U01B-08A UED 21-MAR-79 16:29:50 PAGE 002

```

0012      COMMON /DATAA/ TCC
      END OF DATA DEFINITION
      NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CROPEN
      -----
      *****

```

```

0013      SET SUCCESS
      ITC2=0
      CHECK FOR PSEUDO COLOR MEMORY
      IF((ITCC.EQ.TCUPS).OR.(TCC.EQ.TCRPS)) GO TO 100
      FUNCTION MEMORY
      CORRECT FUNCTION MEMORY NUMBER
      LIMIT FUNCTION MEMORY NUMBER < 4
      NO-IFRNO-1
      IF((NO.GT.MAXGM).OR.(NO.LT.0)) NO=0
      SET FUNCTION MEMORY NUMBER IN TRANSFER COMMAND
      DDB(3,TCC)=NO*400
      GO TO 200
      PSEUDO COLOR MEMORY
      DDB(3,TCC)=0
      SET BUFFER ADDRESS
      DDB(4,TCC)=IADDR(BUFF)
      CALL HANDLER

```

```

C      IC=IURITU(256,DBB(1,TCO),1,ICHIN)
C      CHECK FOR ERROR
C      IF(IC.LT.0) ITC8=IC
C      RETURN
C      END
0023
0024
0025
0027

```

RT-11 FORTRAN IV STORAGE MAP

NAME OFFSET ATTRIBUTES

```

BUFF 000016 INTEGER12 PARAMETER ARRAY (256)
ITC2 000020 INTEGER12 VARIABLE
IFMNO 000014 INTEGER12 PARAMETER VARIABLE
NO 000076 INTEGER12 VARIABLE
IADDR 000000 INTEGER12 PROCEDURE
IC 000100 INTEGER12 VARIABLE
IURITU 000000 INTEGER12 PROCEDURE

```

COMMON BLOCK /DBB/ LENGTH 000500

DBB 000000 INTEGER12 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/ LENGTH 000044

```

TCUIN 000000 INTEGER12 VARIABLE
TCRIN 000002 INTEGER12 VARIABLE
TCUAM 000004 INTEGER12 VARIABLE
TCRGM 000006 INTEGER12 VARIABLE
TCUPS 000010 INTEGER12 VARIABLE
TCRPS 000012 INTEGER12 VARIABLE
TCUAM 000014 INTEGER12 VARIABLE
TCUAM 000016 INTEGER12 VARIABLE
TCUAM 000020 INTEGER12 VARIABLE
TCRTP 000022 INTEGER12 VARIABLE
TCRTP 000024 INTEGER12 VARIABLE
TCRDC 000026 INTEGER12 VARIABLE
TCUAM 000030 INTEGER12 VARIABLE
TCRCS 000032 INTEGER12 VARIABLE
TCUCD 000034 INTEGER12 VARIABLE
TCPCD 000036 INTEGER12 VARIABLE
TCUDB 000040 INTEGER12 VARIABLE
TCRDB 000042 INTEGER12 VARIABLE

```

COMMON BLOCK /CHNN/ LENGTH 000002

ICHIN 000000 INTEGER12 VARIABLE

COMMON BLOCK /COMMON/ LENGTH 000006
 MAIN 000000 INTEGER32 VARIABLE
 MAIN 000002 INTEGER32 VARIABLE
 MAIN 000004 INTEGER32 VARIABLE
 COMMON BLOCK /DATA/ LENGTH 000002
 TCC 000000 INTEGER32 VARIABLE
 1

FILE NAME: COM04.FOR
DATE LAST EDITED: 28-FEB-79

HISTORY:
DATE: 28-FEB-79
MODIFICATION
INITIAL WRITING

INTEGER FUNCTION UTP(IX,IY,IXY)

FUNCTIONAL DESCRIPTION:

THIS ROUTINE WRITES X, OR Y, OR BOTH, TO THE
TARGET CURSOR. (BOTH ACTUALLY REQUIRES 2 TRANSFERS)

PARAMETERS:

```

- = X COORDINATE OF CURSOR (1-256)
- = Y COORDINATE OF CURSOR (1-256)
IX = 0
IY = 1
IXY = 2
- = TRANSFER BOTH X AND Y
- = TRANSFER X ONLY
- = TRANSFER Y ONLY

```

ERRORS:

0	=	SUCCESS
-1	=	WRITE ERROR

XXXXXXXXXXXXXXXXXXXXXXXXXXXX

DATA DEFINITION

INTEGER TCUIN, TCURIN, TCURM, TCUPS, TCAPS
 INTEGER TCUPM, TCURP, TCUDC, TCDC, TCDC
 INTEGER TCUCB, TCRCB, TCUCD, TCRCB, TCRCB

```

INTEGER DDB(8,20)
INTEGER TCC

```

```

0007 C      COMMON /DBB/ DBB
0008 C      COMMON /TC/ TCUIH, TCRH, TCUH, TCRH, TCRH, TCRH, TCRH, TCRH,
      1 TCUIH, TCRH, TCUH, TCRH, TCRH, TCRH, TCRH, TCRH,
      1 TCUIH, TCRH, TCUH, TCRH, TCRH, TCRH, TCRH, TCRH,
0009 C      COMMON /CHAN/ ICHIN

```

RT-11 FORTRAN IV U013-08A WED 28-FEB-79 14:16:39 PAGE 002

```

0010 C      COMMON /CONVAR/ MAXIN, MAXLN, MAXGM
0011 C      COMMON /DATAA/ TCC
      C      END OF DATA DEFINITION
      C      NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CROPER
      C      -----
      C      *****

```

```

0012 C      SET SUCCESS
0013 C      UTP=0
      C      SET FUNCTION -WRITE TARGET POSITION
0014 C      TCC=TCUTP
0015 C      CHECK FOR X,Y,OR BOTH
      C      IF(IYV.EQ.0) GO TO 100
0016 C      IF(IYV.EQ.1) GO TO 200
      C      JUST Y
      C      SET TRANSFER COMMAND
0017 C      DBB(3,TCC)=IV
0018 C      GO TO 50
0019 C
      C      JUST X

```

0020 DB(3,TCC)-IX
0021 GO TO 50

200

CCCC

CALL HANDLER

IC-IURITU(256,DB(1,TCC),1,ICHIN)

50

CHECK FOR ERROR

IF(IC.LT.0) UTP-IC

RETURN

CCCC

0023

0025

CCCC

DB(3,TCC)-IX
IC-IURITU(256,DB(1,TCC),256,1,ICHIN)
DB(3,TCC)-IV

100

0026
0027
0028

PAGE 003

RT-11 FORTRAN IV U018-08A WED 28-FEB-79 14:16:39

0029 ID-IURITU(256,DB(1,TCC),256,1,ICHIN)

C CHECK ERROR

IF(IC.LT.0) UTP-IC

IF(ID.LT.0) UTP-ID

C

C

C

RETURN

END

0034

0035

PT-11 FORTRAN IV STORAGE MAP

NAME OFFSET ATTRIBUTES

UTP	000022	INTEGER12	VARIABLE
IX	000014	INTEGER12	PARAMETER
IV	000016	INTEGER12	PARAMETER
I/V	000020	INTEGER12	PARAMETER
IC	000100	INTEGER12	VARIABLE
IURITU	000000	INTEGER12	PROCEDURE
ID	000102	INTEGER12	VARIABLE

```

COMMON BLOCK /DDB/          LENGTH 000500
DOE  000000  INTEGER12 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/          LENGTH 000044
TCUIM 000000  INTEGER12 VARIABLE
TCRIM 000002  INTEGER12 VARIABLE
TCUGH 000004  INTEGER12 VARIABLE
TCRGM 000006  INTEGER12 VARIABLE
TCUPS 000010  INTEGER12 VARIABLE
TCRPS 000012  INTEGER12 VARIABLE
TCUFM 000014  INTEGER12 VARIABLE
TCRFH 000016  INTEGER12 VARIABLE
TCUTP 000020  INTEGER12 VARIABLE
TCRTP 000022  INTEGER12 VARIABLE
TCUDC 000024  INTEGER12 VARIABLE
TCRDC 000026  INTEGER12 VARIABLE
TCUCB 000030  INTEGER12 VARIABLE
TCRCB 000032  INTEGER12 VARIABLE
TCUCD 000034  INTEGER12 VARIABLE
TCRCD 000036  INTEGER12 VARIABLE
TCUDB 000040  INTEGER12 VARIABLE
TCRDB 000042  INTEGER12 VARIABLE

COMMON BLOCK /CMAN/        LENGTH 000002
ICHIN 000000  INTEGER12 VARIABLE

COMMON BLOCK /COMUAR/      LENGTH 000006
MAXIM 000000  INTEGER12 VARIABLE
MAXLM 000002  INTEGER12 VARIABLE
MAXGM 000004  INTEGER12 VARIABLE

COMMON BLOCK /DATAA/       LENGTH 000002
TCC  000000  INTEGER12 VARIABLE

```

RT-11 FORTRAN IV U01B-08A WED 28-FEB-79 14:16:41 PAGE 00

```

C
C      INTEGER FUNCTION RTP(IX,IY)
C
C      FUNCTIONAL DESCRIPTION:
C
C      THIS ROUTINE READS THE X AND Y POSITION OF THE TARGET
C      AND PUTS THE COORDINATES IN IX, AND IY .
C
C      PARAMETERS:

```



```

00000000000000000000 00
IX : X POSITION OF TARGET (1-256)
IV : Y POSITION OF TARGET (1-256)
ERRORS:
0 - SUCCESS
-1 - WRITE ERROR
*****
DATA DEFINITION
INTEGER TAR(2)
0001
INTEGER TCUIN, TORIN, TOUN, TCRGN, TCUPS, TCRPS
0002
INTEGER TCUFN, TCRFN, TCUTP, TCRTP, TCUDC, TCRDC
0003
INTEGER TCUCB, TCRCB, TCUCD, TCRCD, TCUDB, TCRDB
0004
INTEGER DDB(8,20)
0005
INTEGER TCC
0006
COMMON /DDB/ DDB
0007
COMMON /TC/ TCUIN, TORIN, TOUN, TCRGN, TCUPS, TCRPS,
0008
TCUFN, TCRFN, TCUTP, TCRTP, TCUDC, TCRDC,
TCUCB, TCRCB, TCUCD, TCRCD, TCUDB, TCRDB
COMMON /CHAN/ ICHIN
0009
COMMON /CONUAR/ MAXIN,MAXLN,MAXGN
0010
COMMON /DATAA/ TCC
0011
END OF DATA DEFINITION
NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CROPER
-----
*****
SET SUCCESS
*****

```

```

C C C
0013 SET FUNCTION = READ TARGET POSITION
TCC=TCRTP
C C C
0014 GET ADDRESS
DDB(4,TCC)=IADDR(TAR)
C C C
0015 CALL HANDLER
IC=IURITU(256,DDB(1,TCC),1,ICHIN)
C C C
0016 CHECK FOR ERRORS
IF(IC.LT.0)RTP=IC
C C C
0018 SET X, AND Y (4097 IS OFFSET FROM 1)
IX=TAR(1)+4097
IY=TAR(2)+4097
C
0020 RETURN
0021 END

```

PT-11 FORTRAN IV STORAGE MAP

NAME	OFFSET	ATTRIBUTES	LENGTH
TAP	000006	INTEGER12 ARRAY (2)	
PTP	000006	REAL84 VARIABLE	
IADDR	000000	INTEGER12 PROCEDURE	
IC	000072	INTEGER12 VARIABLE	
IURITU	000000	INTEGER12 PROCEDURE	
I/	000074	INTEGER12 VARIABLE	
IY	000076	INTEGER12 VARIABLE	
COMMON BLOCK /DDB/ LENGTH 000500			
DDB	000000	INTEGER12 ARRAY (8,20) VECTORED	
COMMON BLOCK /TC/ LENGTH 000044			
TCUIM	000000	INTEGER12 VARIABLE	
TCPIH	000002	INTEGER12 VARIABLE	
TCUGM	000004	INTEGER12 VARIABLE	
TCPCM	000006	INTEGER12 VARIABLE	
TCUPS	000010	INTEGER12 VARIABLE	
TCPPS	000012	INTEGER12 VARIABLE	
TCUFM	000014	INTEGER12 VARIABLE	
TCRFM	000016	INTEGER12 VARIABLE	

```

TCUTP 000020 INTEGER12 VARIABLE
TCUTN 000022 INTEGER12 VARIABLE
TCUN2 000024 INTEGER12 VARIABLE
TCUN3 000026 INTEGER12 VARIABLE
TCURC 000030 INTEGER12 VARIABLE
TCURC 000032 INTEGER12 VARIABLE
TCURC 000034 INTEGER12 VARIABLE
TCURC 000036 INTEGER12 VARIABLE
TCURC 000040 INTEGER12 VARIABLE
TCURC 000042 INTEGER12 VARIABLE
COMMON BLOCK /CHAN/ LENGTH 000002
ICIN 000000 INTEGER12 VARIABLE
COMMON BLOCK /CONVAR/ LENGTH 000006
MAXIN 000000 INTEGER12 VARIABLE
MAXIN 000002 INTEGER12 VARIABLE
MAXIN 000004 INTEGER12 VARIABLE
COMMON BLOCK /DATAA/ LENGTH 000002
TCC 000000 INTEGER12 VARIABLE

```

RT-11 FORTRAN IU U018-08A UED 18-APR-79 16:30:03

FILE NAME: COMLOS.FOR

DATE LAST EDITED: 1-MAR-79

HISTORY:

DATE: MODIFICATION

1-MAR-79 INITIAL WRITING

INTEGER FUNCTION IUCB(BUFF)

FUNCTIONAL DESCRIPTION:

THIS ROUTINE IS CALLED TO TRANSMIT A BLOCK OF TEXT TO THE TEXT BUFFER IN THE CORTAL. AFTER THE TEXT IS RECEIVED, THE CORTAL WILL USE THIS TEXT AS IF IT WERE TYPED FROM THE KEYBOARD.

PARAMETERS:

BUFF - USER BUFFER OF TEXT

ERRORS:

0 - SUCCESS
-1 - WRITE ERROR

DATA DEFINITION

LOGICAL*1 BUFF(512)

INTEGER TCUM, TCRIM, TCUMH, TCRGH, TCUPS, TCPS
INTEGER TCUMF, TCRFH, TCUTP, TCUDC, TCDC
INTEGER TCUCB, TCRCB, TCUCD, TCRCD, TCADB, TCADB

INTEGER DB(8,20)
INTEGER TC

0001

0002

0003

0004

0005

0006

0007

0001

0002

0003

0004

0005

0006

0007

PT-1: FORTRAN IU U019-08A WED 18-APR-79 16:30:03 PAGE 002

```

C      ;      1) SET COMMAND
C      ;      2) CALL COMMON CODE FOR TRANSFER CODE 3 TYPE COMMANDS
C
C      TCC-TCUCB
C      0013
C      IUCB-ITC3(BUFF, IADR)
C      0014
C      RETURN
C      0015
C      END
C      0016

```

FT-11 FORTRAN IV		STORAGE MAP
NAME	OFFSET	ATTRIBUTES
BUFF	000014	LOCAL#1 PARAMETER ARRAY (512)
LOCB	000016	INTEGER#1 VARIABLE
LOC	000018	INTEGER#2 VARIABLE
LOC2	000020	INTEGER#2 VARIABLE
LOC3	000022	INTEGER#2 VARIABLE
LOC4	000024	INTEGER#2 VARIABLE
LOC5	000026	INTEGER#2 VARIABLE
LOC6	000028	INTEGER#2 VARIABLE
LOC7	000030	INTEGER#2 VARIABLE
LOC8	000032	INTEGER#2 VARIABLE
LOC9	000034	INTEGER#2 VARIABLE
LOC10	000036	INTEGER#2 VARIABLE
LOC11	000038	INTEGER#2 VARIABLE
LOC12	000040	INTEGER#2 VARIABLE
LOC13	000042	INTEGER#2 VARIABLE
LOC14	000044	INTEGER#2 VARIABLE
LOC15	000046	INTEGER#2 VARIABLE
LOC16	000048	INTEGER#2 VARIABLE
LOC17	000050	INTEGER#2 VARIABLE
LOC18	000052	INTEGER#2 VARIABLE
LOC19	000054	INTEGER#2 VARIABLE
LOC20	000056	INTEGER#2 VARIABLE
LOC21	000058	INTEGER#2 VARIABLE
LOC22	000060	INTEGER#2 VARIABLE
LOC23	000062	INTEGER#2 VARIABLE
LOC24	000064	INTEGER#2 VARIABLE
LOC25	000066	INTEGER#2 VARIABLE
LOC26	000068	INTEGER#2 VARIABLE
LOC27	000070	INTEGER#2 VARIABLE
LOC28	000072	INTEGER#2 VARIABLE
LOC29	000074	INTEGER#2 VARIABLE
LOC30	000076	INTEGER#2 VARIABLE
LOC31	000078	INTEGER#2 VARIABLE
LOC32	000080	INTEGER#2 VARIABLE
LOC33	000082	INTEGER#2 VARIABLE
LOC34	000084	INTEGER#2 VARIABLE
LOC35	000086	INTEGER#2 VARIABLE
LOC36	000088	INTEGER#2 VARIABLE
LOC37	000090	INTEGER#2 VARIABLE
LOC38	000092	INTEGER#2 VARIABLE
LOC39	000094	INTEGER#2 VARIABLE
LOC40	000096	INTEGER#2 VARIABLE
LOC41	000098	INTEGER#2 VARIABLE
LOC42	000100	INTEGER#2 VARIABLE
LOC43	000102	INTEGER#2 VARIABLE
LOC44	000104	INTEGER#2 VARIABLE
LOC45	000106	INTEGER#2 VARIABLE
LOC46	000108	INTEGER#2 VARIABLE
LOC47	000110	INTEGER#2 VARIABLE
LOC48	000112	INTEGER#2 VARIABLE
LOC49	000114	INTEGER#2 VARIABLE
LOC50	000116	INTEGER#2 VARIABLE
LOC51	000118	INTEGER#2 VARIABLE
LOC52	000120	INTEGER#2 VARIABLE
LOC53	000122	INTEGER#2 VARIABLE
LOC54	000124	INTEGER#2 VARIABLE
LOC55	000126	INTEGER#2 VARIABLE
LOC56	000128	INTEGER#2 VARIABLE
LOC57	000130	INTEGER#2 VARIABLE
LOC58	000132	INTEGER#2 VARIABLE
LOC59	000134	INTEGER#2 VARIABLE
LOC60	000136	INTEGER#2 VARIABLE
LOC61	000138	INTEGER#2 VARIABLE
LOC62	000140	INTEGER#2 VARIABLE
LOC63	000142	INTEGER#2 VARIABLE
LOC64	000144	INTEGER#2 VARIABLE
LOC65	000146	INTEGER#2 VARIABLE
LOC66	000148	INTEGER#2 VARIABLE
LOC67	000150	INTEGER#2 VARIABLE
LOC68	000152	INTEGER#2 VARIABLE
LOC69	000154	INTEGER#2 VARIABLE
LOC70	000156	INTEGER#2 VARIABLE
LOC71	000158	INTEGER#2 VARIABLE
LOC72	000160	INTEGER#2 VARIABLE
LOC73	000162	INTEGER#2 VARIABLE
LOC74	000164	INTEGER#2 VARIABLE
LOC75	000166	INTEGER#2 VARIABLE
LOC76	000168	INTEGER#2 VARIABLE
LOC77	000170	INTEGER#2 VARIABLE
LOC78	000172	INTEGER#2 VARIABLE
LOC79	000174	INTEGER#2 VARIABLE
LOC80	000176	INTEGER#2 VARIABLE
LOC81	000178	INTEGER#2 VARIABLE
LOC82	000180	INTEGER#2 VARIABLE
LOC83	000182	INTEGER#2 VARIABLE
LOC84	000184	INTEGER#2 VARIABLE
LOC85	000186	INTEGER#2 VARIABLE
LOC86	000188	INTEGER#2 VARIABLE
LOC87	000190	INTEGER#2 VARIABLE
LOC88	000192	INTEGER#2 VARIABLE
LOC89	000194	INTEGER#2 VARIABLE
LOC90	000196	INTEGER#2 VARIABLE
LOC91	000198	INTEGER#2 VARIABLE
LOC92	000200	INTEGER#2 VARIABLE
LOC93	000202	INTEGER#2 VARIABLE
LOC94	000204	INTEGER#2 VARIABLE
LOC95	000206	INTEGER#2 VARIABLE
LOC96	000208	INTEGER#2 VARIABLE
LOC97	000210	INTEGER#2 VARIABLE
LOC98	000212	INTEGER#2 VARIABLE
LOC99	000214	INTEGER#2 VARIABLE
LOC100	000216	INTEGER#2 VARIABLE
LOC101	000218	INTEGER#2 VARIABLE
LOC102	000220	INTEGER#2 VARIABLE
LOC103	000222	INTEGER#2 VARIABLE
LOC104	000224	INTEGER#2 VARIABLE

```

COMMON BLOCK /DB/      LENGTH 000000
DB      000000  INTEGER*2 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/      LENGTH 000044
TCUIN  000000  INTEGER*2 VARIABLE
TCRIN  000002  INTEGER*2 VARIABLE
TCUGM  000004  INTEGER*2 VARIABLE
TCRGM  000006  INTEGER*2 VARIABLE
TCUPS  000010  INTEGER*2 VARIABLE
TCRPS  000012  INTEGER*2 VARIABLE
TCUFM  000014  INTEGER*2 VARIABLE
TCRFH  000016  INTEGER*2 VARIABLE
TCUTP  000020  INTEGER*2 VARIABLE
TCRTP  000022  INTEGER*2 VARIABLE
TCUDC  000024  INTEGER*2 VARIABLE
TCRDC  000026  INTEGER*2 VARIABLE
TCUCB  000030  INTEGER*2 VARIABLE
TCRCB  000032  INTEGER*2 VARIABLE
TCUCD  000034  INTEGER*2 VARIABLE
TCRCD  000036  INTEGER*2 VARIABLE
TCUDB  000040  INTEGER*2 VARIABLE
TCRDB  000042  INTEGER*2 VARIABLE

COMMON BLOCK /CHAN/    LENGTH 000002
ICHIN  000000  INTEGER*2 VARIABLE

COMMON BLOCK /COMVAR/  LENGTH 000006
MAXIM  000000  INTEGER*2 VARIABLE
MAXLN  000002  INTEGER*2 VARIABLE
MAXGM  000004  INTEGER*2 VARIABLE

COMMON BLOCK /DATA/    LENGTH 000002
TCC      000000  INTEGER*2 VARIABLE

```

RT-11 FORTRAN IV U01B-08A WED 18-APR-79 16:30:08 PAGE 001

```

0001  C      INTEGER FUNCTION IRCB(BUFF)
      C      FUNCTIONAL DESCRIPTION:
      C      THIS ROUTINE IS CALLED TO READ THE BLOCK OF TEXT
      C      FROM THE BUFFER IN THE CONTAL. THIS WOULD BE USED
      C      WHEN THE CONTAL INTERRUPTED VIA THE 'SEND MESSAGE' REQUEST,
      C      TO READ THE TEXT TO DETERMINE WHAT THE MESSAGE WAS.

```



```

      000000 INTEGER12 VARIABLE
      000002 INTEGER12 VARIABLE
      000004 INTEGER12 VARIABLE
      COMMON BLOCK DATA  LENGTH 000002
      TCC 000000 INTEGER12 VARIABLE

```

RT-11 FORTRAN IV U01B-08A WED 18-APR-79 16:30:13 PAGE 001

```

0001 C INTEGER FUNCTION UDB(BUFF)

```

FUNCTIONAL DESCRIPTION:

THIS ROUTINE IS CALLED TO TRANSMIT THE DATA BLOCK TO THE COMTEL. THE DATA BLOCK CONTAINS THE CONTENTS OF SEVERAL DISPLAY REGISTERS THAT DETERMINE THE STATUS OF THE DISPLAY.

PARAMETERS:

BUFF = USER BUFFER OF TEXT

ERRORS:

0 = SUCCESS
-1 = WRITE ERROR

DATA DEFINITION

LOGICAL*1 BUFF(S12)

INTEGER TCUM, TCRM, TCUH, TCRG, TCUR, TCURP, TCURD, TCURC
INTEGER TCUM, TCRM, TCUH, TCRG, TCUR, TCURP, TCURD, TCURC
INTEGER TCUB, TCUR, TCURD, TCURC, TCURD, TCURC, TCURD, TCURC

INTEGER DDB(8,20)
INTEGER TCC

COMMON /DDB/ DDB

0002 C

0003 C
0004 C
0005 C

0006 C
0007 C

0008 C

```

C      COMMON /TC/ TCUIR, TCURR, TCURH, TCURH, TCURP, TCURP, TCURD,
1      TCURF, TCURF, TCURF, TCURF, TCURF, TCURF, TCURF, TCURF,
1      TCURB, TCURB, TCURB, TCURB, TCURB, TCURB, TCURB, TCURB
C      COMMON /CHAN/ ICHIN
C      COMMON /CONVAR/ MAXIN, MAXLN, MAXGM
C      COMMON /DATA/ TCC
C      END OF DATA DEFINITION
C      NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CMOPEN
C      -----
C      *****

```

PAGE 002

RT-11 FORTRAN IV U01B-02A UED 18-APR-79 16:30:13

```

C      1) SET COMMAND
C      2) CALL COMMON CODE FOR TRANSFER CODE 3 TYPE COMMANDS
C      TCC-TCURB
0013 C      UDB-ITC3(BUFF, IADR)
0014 C      RETURN
0015 C      END
0016

```

```

RT-11 FORTRAN IV STORAGE MAP
NAME OFFSET ATTRIBUTES
BUFF 000014 LOGICALX11 PARAMETER ARRAY (512)
UDE 000016 INTEGERX2 VARIABLE
ITC3 000000 INTEGERX2 PROCEDURE
IADR 000070 INTEGERX2 VARIABLE
COMMON BLOCK /DDB/ LENGTH 000500
DDB 000000 INTEGERX2 ARRAY (8,20) VECTORED

```

```

COMMON BLOCK /TC/          LENGTH 000044
TCUIN 000000 INTEGER12 VARIABLE
TCUIM 000002 INTEGER12 VARIABLE
TCUJM 000004 INTEGER12 VARIABLE
TCUJM 000006 INTEGER12 VARIABLE
TCUJS 000010 INTEGER12 VARIABLE
TCUJS 000012 INTEGER12 VARIABLE
TCUJS 000014 INTEGER12 VARIABLE
TCUJS 000016 INTEGER12 VARIABLE
TCUJS 000020 INTEGER12 VARIABLE
TCUJS 000022 INTEGER12 VARIABLE
TCUJS 000024 INTEGER12 VARIABLE
TCUJS 000026 INTEGER12 VARIABLE
TCUJS 000030 INTEGER12 VARIABLE
TCUJS 000032 INTEGER12 VARIABLE
TCUJS 000034 INTEGER12 VARIABLE
TCUJS 000036 INTEGER12 VARIABLE
TCUJS 000040 INTEGER12 VARIABLE
TCUJS 000042 INTEGER12 VARIABLE
COMMON BLOCK /CHAN/        LENGTH 000002
ICHIN 000000 INTEGER12 VARIABLE
COMMON BLOCK /COMVAR/      LENGTH 000006
MAXIN 000000 INTEGER12 VARIABLE
MAXIN 000002 INTEGER12 VARIABLE
MAXIN 000004 INTEGER12 VARIABLE
COMMON BLOCK /DATAA/       LENGTH 000002
TCC 000000 INTEGER12 VARIABLE

```

PT-11 FORTRAN IV U018-08A UED 18-APR-79 16:30:18 PAGE 001

```

0001 C INTEGER FUNCTION RDB(BUFF)
C C
C C FUNCTIONAL DESCRIPTION:
C C
C C THIS ROUTINE IS CALLED TO READ THE CORTAL DATA BLOCK.
C C THE DATA BLOCK CONTAINS THE TARGET POSITION, AND THE
C C DISPLAY REGISTERS.
C C
C C PARAMETERS:

```

```

00000000000000000000 00
0002
0003
0004
0005
0006
0007
0008
0009
0010
0011
0012
00000000000000000000 00

BUFF - USER BUFFER OF TEXT
ERRORS:
0 - SUCCESS
-1 - UNITE ERROR

*****
DATA DEFINITION
LOGICAL*1 BUFF(512)

INTEGER TCUM, TCRIM, TCUMN, TCRGM, TCUPS, TCRPS
INTEGER TCUMN, TCRFN, TCUTP, TCRTD, TCUDC, TCRDC
INTEGER TCUCB, TCRCS, TCUCD, TCRCD, TCUDB, TCRDB

INTEGER DB(8,20)
INTEGER TCC

COMMON /DB/ DB

COMMON /TC/ TCUM, TCRIM, TCUMN, TCRGM, TCUPS, TCRPS,
TCUMN, TCRFN, TCUTP, TCRTD, TCUDC, TCRDC,
TCUCB, TCRCS, TCUCD, TCRCD, TCUDB, TCRDB

COMMON /CHAN/ ICHIN

COMMON /COMVAR/ MAXIN, MAXLN, MAXGM

COMMON /DATA/ TCC

END OF DATA DEFINITION

NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CROPEN
-----
*****

```

00000
1) SET COMMAND
2) CALL COMMON CODE FOR TRANSFER CODE 3 TYPE COMMANDS

```

C      TCC-TCRDB
0213 C
0214 C   RDB-ITC3(BUFF,IADR)
C
0215 C   RETURN
0216 C   END

```

```

RT-11 FORTRAN IV      STORAGE MAP

NAME  OFFSET  ATTRIBUTES
RUFF  000014  LOGICAL111 PARAMETER ARRAY (512)
RDB   000016  INTEGER12 VARIABLE
ITC3  000000  INTEGER12 PROCEDURE
IADR  000070  INTEGER12 VARIABLE

COMMON BLOCK /RDB/      LENGTH 000500

DDB   000000  INTEGER12 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/      LENGTH 000044
TCUIM 000000  INTEGER12 VARIABLE
TCRIM 000002  INTEGER12 VARIABLE
TCUGM 000004  INTEGER12 VARIABLE
TCRGM 000006  INTEGER12 VARIABLE
TCUPS 000010  INTEGER12 VARIABLE
TCRPS 000012  INTEGER12 VARIABLE
TCUFM 000014  INTEGER12 VARIABLE
TCRFM 000016  INTEGER12 VARIABLE
TCUTP 000020  INTEGER12 VARIABLE
TCRTP 000022  INTEGER12 VARIABLE
TCUDC 000024  INTEGER12 VARIABLE
TCPDC 000026  INTEGER12 VARIABLE
TCUCB 000030  INTEGER12 VARIABLE
TCRCB 000032  INTEGER12 VARIABLE
TCUCD 000034  INTEGER12 VARIABLE
TCRCD 000036  INTEGER12 VARIABLE
TCUDB 000040  INTEGER12 VARIABLE
TCRDB 000042  INTEGER12 VARIABLE

COMMON BLOCK /CHAN/    LENGTH 000002
ICHIN 000000  INTEGER12 VARIABLE

COMMON BLOCK /COMUAR/  LENGTH 000006
MAXIM 000000  INTEGER12 VARIABLE
MAXLM 000002  INTEGER12 VARIABLE
MAXGM 000004  INTEGER12 VARIABLE

```

COMMON BLOCK /DATA/ LENGTH 000002
 TCC 000000 INTEGER12 VARIABLE

RT-11 FORTRAN IV U018-08A UED 18-APR-79 16:30:22 PAGE 001

0001 INTEGER FUNCTION IUCD(BUFF,IAD)

FUNCTIONAL DESCRIPTION:

THIS ROUTINE IS USED TO WRITE TO ANY BLOCK OF MEMORY
 IN THE CORTAL. ITS PRIMARY USE IS IN DOWN LOADING CODE
 TO THE CORTAL, HOWEVER ANY OF THE CORTAL MEMORY LOC.
 CAN BE READ, AND ANY OF THE RAM AREAS WRITTEN.
 THE ADDRESS MUST BE ON A 256 WORD BOUNDARY.

PARAMETERS:

BUFF - USER BUFFER OF TEXT
 IAD - ADDRESS IN THE CORTAL FOR THE TRANSFER
 (MUST BE A 512 BYTE BOUNDARY)

ERRORS:

0 - SUCCESS
 -1 - WRITE ERROR

DATA DEFINITION

LOGICAL*1 BUFF(512)

INTEGER TCUM, TCRIM, TCUGM, TCRGM, TCUPS, TCRPS
 INTEGER TCUFH, TCRFH, TCUTP, TCRTP, TCUDC, TCRDC
 INTEGER TCUCB, TCRCB, TCUCD, TCRCD, TCUDB, TCRDB

INTEGER DDB(8,20)
 INTEGER TCC

COMMON /DDB/ DDB

0003

NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CROPEN

1) SET COMMAND
2) SET COMMON
3) SET COMMON CODE FOR TRANSFER CODE 3 TYPE COMMANDS

IUCD-ITC3(BUFF, IADR)

COMMUNITY ADULT / BDB / LENGTH 000500

```
DDN 000000 INTEGER12 ARRAY (8,20) VECTORED
```

```

COMMON BLOCK /TC/      LENGTH 000044
TCMIM 000000 INTEGER12 VARIABLE
TCRIM 000002 INTEGER12 VARIABLE
TCUGH 000004 INTEGER12 VARIABLE
TCRGM 000006 INTEGER12 VARIABLE
TCRPS 000010 INTEGER12 VARIABLE
TCRPS 000012 INTEGER12 VARIABLE
TCUFM 000014 INTEGER12 VARIABLE
TCRFM 000016 INTEGER12 VARIABLE
TCWTP 000020 INTEGER12 VARIABLE
TCWTP 000022 INTEGER12 VARIABLE
TCUDC 000024 INTEGER12 VARIABLE
TCUDC 000026 INTEGER12 VARIABLE
TCUCB 000030 INTEGER12 VARIABLE
TCRCB 000032 INTEGER12 VARIABLE
TCUCD 000034 INTEGER12 VARIABLE
TCRCD 000036 INTEGER12 VARIABLE
TCUDB 000040 INTEGER12 VARIABLE
TCRDB 000042 INTEGER12 VARIABLE

COMMON BLOCK /CHAN/    LENGTH 000002
ICHIN 000000 INTEGER12 VARIABLE

COMMON BLOCK /CONVAR/  LENGTH 000006
MAXIM 000000 INTEGER12 VARIABLE
MAXLN 000002 INTEGER12 VARIABLE
MAXGM 000004 INTEGER12 VARIABLE

COMMON BLOCK /DATA/    LENGTH 000002
TCC 000000 INTEGER12 VARIABLE

```

PT-11 FORTRAN IV U018-08A UED 18-APR-79 16:30:26 PAGE 001

```

C
0001 C INTEGER FUNCTION IRCD(BUFF,IAD)
C
C FUNCTIONAL DESCRIPTION:
C
C THIS ROUTINE IS USED TO READ FROM ANY BLOCK OF MEMORY
C IN THE CORTAL. THE ADDRESS MUST BE ON A 256 WORD
C BOUNDARY.
C
C PARAMETERS:

```


2) CALL COMMON CODE FOR TRANSFER CODE 3 TYPE COMMANDS

```

0013 C TCC=TCRCD
0014 C IRCD=ITC3(BUFF,IADR)
0015 C RETURN
0016 C END

```

PT-11 FORTRAN IV STORAGE MAP

NAME	OFFSET	ATTRIBUTES
BUFF	000014	LOGICAL*11 PARAMETER ARRAY (512)
IRCD	000020	INTEGER*2 VARIABLE
IAD	000016	INTEGER*2 PARAMETER VARIABLE
ITC3	000000	INTEGER*2 PROCEDURE
IADR	000072	INTEGER*2 VARIABLE
COMMON BLOCK /DDB/ LENGTH 000500		
DDB	000000	INTEGER*2 ARRAY (8,20) VECTORED
COMMON BLOCK /TC/ LENGTH 000044		
TCUIM	000000	INTEGER*2 VARIABLE
TCRIM	000002	INTEGER*2 VARIABLE
TCUGM	000004	INTEGER*2 VARIABLE
TCRGM	000006	INTEGER*2 VARIABLE
TCUPS	000010	INTEGER*2 VARIABLE
TCPPS	000012	INTEGER*2 VARIABLE
TCUFM	000014	INTEGER*2 VARIABLE
TCRFM	000016	INTEGER*2 VARIABLE
TCUTP	000020	INTEGER*2 VARIABLE
TCRTP	000022	INTEGER*2 VARIABLE
TCUDC	000024	INTEGER*2 VARIABLE
TCRDC	000026	INTEGER*2 VARIABLE
TCUCB	000030	INTEGER*2 VARIABLE
TCPCB	000032	INTEGER*2 VARIABLE
TCUCD	000034	INTEGER*2 VARIABLE
TCRCD	000036	INTEGER*2 VARIABLE
TCUDB	000040	INTEGER*2 VARIABLE
TCFDB	000042	INTEGER*2 VARIABLE
COMMON BLOCK /CHAN/ LENGTH 000002		
ICWIN	000000	INTEGER*2 VARIABLE
COMMON BLOCK /COMUAR/ LENGTH 000006		
MAXIM	000000	INTEGER*2 VARIABLE

```

      000002 INTEGER*2 VARIABLE
      000004 INTEGER*2 VARIABLE
      COMMON BLOCK /DATA/  LENGTH 000002
      TCC 000000 INTEGER*2 VARIABLE

```

RT-11 FORTRAN IV U01B-08A UED 18-APR-79 16:30:31 PAGE 001

```

0001 C      INTEGER FUNCTION ITC3(BUFF,IAD)
      C      FUNCTIONAL DESCRIPTION:
      C      THIS IS THE COMMON ROUTINE FOR TRANSFER CODE 3
      C      TYPE COMMANDS. IT IS CALLED BY READ/WRITE - COMMAND
      C      BLOCK, CODE DATA, AND DATA BLOCK.
      C      PARAMETERS:
      C      BUFF - USER BUFFER OF TEXT
      C      IAD - ADDRESS IN THE CORTAL FOR THE TRANSFER
      C      ERRORS:
      C      0 - SUCCESS
      C      -1 - WRITE ERROR
      C      *****
      C      DATA DEFINITION
      C      LOGICAL*1 BUFF(512)
      C      INTEGER TCUIH, TCRIH, TCUGH, TCRGH, TCUPS, TCRPS
      C      INTEGER TCURF, TCRRF, TCURP, TCRTP, TCUDC, TCRDC
      C      INTEGER TCUCB, TCRCB, TCUCD, TCRCD, TCUCB, TCRDB
      C      INTEGER DDB(8,20)
      C      INTEGER TCC
      C      COMMON /DDB/ DDB
      C

```

1-117

```

RT-11 FORTRAN IV      STORAGE MAP

NAME  OFFSET  ATTRIBUTES

BUFF  000014  LOGICAL11  PARAMETER  ARRAY (512)
ITC3  000020  INTEGER12 VARIABLE
IAD   000016  INTEGER12 PARAMETER  VARIABLE
IA    000076  INTEGER12 VARIABLE
IADDR 000000  INTEGER12 PROCEDURE
IC    000100  INTEGER12 VARIABLE
ICUTU 000000  INTEGER12 PROCEDURE

COMMON BLOCK /DDB/      LENGTH 000500

DDB   000000  INTEGER12 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/      LENGTH 000044

TCUIM 000000  INTEGER12 VARIABLE
TCRIM 000002  INTEGER12 VARIABLE
TCUGM 000004  INTEGER12 VARIABLE
TCRGM 000006  INTEGER12 VARIABLE
TCUPS 000010  INTEGER12 VARIABLE
TCRPS 000012  INTEGER12 VARIABLE
TCUFM 000014  INTEGER12 VARIABLE
TCRFM 000016  INTEGER12 VARIABLE
TCUTP 000020  INTEGER12 VARIABLE
TCRTP 000022  INTEGER12 VARIABLE
TCUDC 000024  INTEGER12 VARIABLE
TCRDC 000026  INTEGER12 VARIABLE
TCUCB 000030  INTEGER12 VARIABLE
TCRCB 000032  INTEGER12 VARIABLE
TCUCD 000034  INTEGER12 VARIABLE
TCRCD 000036  INTEGER12 VARIABLE
TCUDB 000040  INTEGER12 VARIABLE
TCRDB 000042  INTEGER12 VARIABLE

COMMON BLOCK /CHAN/    LENGTH 000002

ICHIM 000000  INTEGER12 VARIABLE

COMMON BLOCK /COMVAR/  LENGTH 000006

MAXIM 000000  INTEGER12 VARIABLE
MAXIM 000002  INTEGER12 VARIABLE
MAXIM 000004  INTEGER12 VARIABLE

COMMON BLOCK /DATAA/   LENGTH 000002

TCC   000000  INTEGER12 VARIABLE
*
```

2.0 INTEGRATION OF THE COMTAL VISION ONE WITH THE RIPS (RADAR) IMAGE PROCESSING SYSTEM

The PAR RIPS system was modified to incorporate a new option "frame", which allows a RIPS user to write and read information in RIPS format to and from the COMTAL Vision One display system. Mechanisms for adding functions to the RIPS system had been built in and documented by PAR, simplifying this effort by Mc².

The approach taken was to modify the COMTAL communication facilities provided by Mc²'s previous COMPIP subsystem to run under RSX-11M, and then to integrate this package with RIPS. Implicit in this effort was the thorough checkout of the COMTAL-supplied RSX-11M device driver, purchased under this contract. This handler is essentially a DEC DMA driver, modified by COMTAL to support their hardware.

In addition to adding the COMTAL to RIPS as a valid input/output device, this sub-effort has produced a set of routines that can be used by any FORTRAN programmer for communicating with the COMTAL under RSX-11M. Evidence of this is the effort of Section III following, where the SRC SILHOUETTE program, written in FORTRAN under RT-11, was modified to use the COMTAL under RSX-11M.

Instructions for altering the RIPS subsystem to accommodate another "menu" of user options were included in PAR's RIPS documentation. Stated briefly, these steps are:

- o Update RIPS file FRAMES.SRC, which contains the text and control transfer codes of the screen menus appearing in RIPS.
- o Run the FRAMES program to convert the above file into its RIPS-usable form, FRAMES.IPS.
- o Add control transfer code to RIPS module CALSUB. Re-compile CALSUB.

- o Update the Overlay Descriptor Language File EXEC.ODL to incorporate any new Mc² software being added to RIPS.
- o Compile or assemble new code to be incorporated in the modified RIPS system.
- o Run the indirect command file RIPS.LNK to create the new version of RIPS.

The accompanying listings show the code either written or modified by Mc² to add the COMTAL to RIPS.

2.1 FRAMES.SRC and FRAMES.IPS

Modifications:

A new RIPS "FRAME" or option menu was added to FRAMES.SRC to allow user selection of one of the nine COMTAL read/write functions added to RIPS. The following excerpt from FRAMES.SRC is annotated to indicate the code added by Mc².

```
.
.
12
F
0
UTILITY FUNCTIONS
RSX$
LOG$
CALL USER PROGRAMS
GENETATE TEST IMAGES
CREATE TEST IMAGES
EXAMINE HEADER TEXTS
LIST IMAGE GREY VALUES
DISPLAY OPTIONS$
MASTER FRAMES
11
0
1
0
2
0
3
0
4
0
5
0
6
0
7
0
8
0
9
F
12
F
0
CONTAL DISPLAY AND COPY OPTIONS$
DISPLAY IMAGE FILES
COPY IMAGE FILE FROM CONTAL$
DISPLAY GRAPHICS FILES
COPY GRAPHICS FILE FROM CONTAL$
DISPLAY FUNCTION FILES
COPY FUNCTION FILE FROM CONTAL$
DISPLAY PSEUDOCOLOR FILES
COPY PSEUDOCOLOR FILE FROM CONTAL$
WRITE COMMAND BLOCK TO CONTAL$
DISPLAY OPTIONS$
MASTER FRAMES
(end of FRAMES.SRC)
```

} Added by Mc²

2.2 CALSUB.FTN

Modifications:

- o Computed "GO TO" statement, which is first executable statement of CALSUB, has the argument after 13 changed from a 21 (signifying option not implemented) to a 14.
- o Code beginning one line before statement 14 and ending at the RETURN statement is added. These lines call the proper Mc^2 - written routine corresponding to the COMTAL option selected by the user from frame 14.

2.3 EXEC.ODL

Modifications:

- o Lines L34 - L39 are added, defining the residency requirements of all new code added by Mc^2 to support the COMTAL.

2.4 CPIPSB.FTN

Authors:

Written for RT-11 by R. Kellogg, Measurement Concept Corp., 1979.

Adapted for RSX-11M by J. Tims, Measurement Concept Corp., 1979.

CPIPSB can be thought of as the interface between Mc^2 software and the RIPS environment. CPIPSB provides the RIPS system access to Mc^2 software, while at the same time, insuring that all requests for normal RIPS I/O services related to the COMTAL requests are passed through RIPS and RIPS's logging services. It is in actuality, a RIPS subroutine, written and added by Mc^2 . Other Mc^2 COMTAL software

is independent of all RIPS conventions.

CPIPSB contains nine subroutines, one corresponding to each of the RIPS frame 14 COMTAL options.

2.5 COMLØ1.FOR

Authors:

Written for RT-11 by R. Kellogg, Measurement Concept Corp., 1979.

Modified for RSX-11M by J. Tims, Measurement Concept Corp., 1979.

COMLØ1 is a data definition block with no executable statements. It provides data element definition information, data equivalence information, and COMTAL function code arrays for passing to IWRITW, which performs the actual RSX-11M QIO\$ operation.

Information in COMLØ1 is used by COMLØ2, COMLØ3, COMLØ4 and COMLØ5.

2.6 COMLØ2.FOR

Authors:

Written for RT-11 by R. Kellogg, Measurement Concept Corp., 1979.

Modified for RSX-11M by J. Tims, Measurement Concept Corp., 1979.

COMLØ2 contains routines to support requests for

- o Read/Write Image Memory
- o Read/Write Graphic Memory

issued through CPIPSB (see Section 1.5).

2.7 COMLØ3.FOR

Authors:

Written for RT-11 by R. Kellogg, Measurement Concept Corp., 1979.

Modified for RSX-11M by J. Tims, Measurement Concept Corp., 1979.

COMLØ3 contains routines to support requests for

- o Read/Write Pseudocolor Memory
- o Read/Write Graphic Memory

Issued through CPIPSB (see Section 1.6).

2.8 COMLØ4.FOR

Authors:

Written for RT-11 by R. Kellogg, Measurement Concept Corp., 1979.

Modified for RSX-11M by J. Tims, Measurement Concept Corp., 1979.

COMLØ4.FOR is not used by the RIPS system. It consists of routines to

- o Read/Write COMTAL target cursor

and hence has no current function under RIPS (see Section 1.7).

2.9 COMLØ5.FOR

Authors:

Written for RT-11 by R. Kellogg, Measurement Concept Corp., 1979.

Modified for RSX-11M by J. Tims, Measurement Concept Corp., 1979.

COMLØ5 contains routines to support requests for

- o Xmit block of text to COMTAL

This function is used in RIPS to format commands to the COMTAL to be interpreted by its MACRO interpreter.

Although this is the only RIPS function supported, COMLØ5 contains many other capabilities not used by RIPS (see Section 1.8).

- o Read block of text from COMTAL
- o Xmit data block to COMTAL
- o Read data block from COMTAL
- o Write block of PDP 11/40 memory to COMTAL (to be used when additional LSI-11 memory is purchased for down-line loading of COMTAL programs)
- o Read block of COMTAL LSI-11 memory from the PDP-11/40.

2.10 IWRITW.MAC

Author:

J. Tims, Measurement Concept Corp., 1979.

IWRITW was written to mimic the RT-11 I/O function IWRITW, which was used by COMPIP. It also includes code to mimic the RT-11 function routine IADDR, which would return the starting address of a FORTRAN parameter. No such routine exists within RSX-11M FORTRAN.

IWRITW accepts I/O requests that "look like" RT-11 requests and converts them into RSX-11M QIOW\$ requests.

Also included in IWRITW.MAC is subroutine IBYTEX, which is used to swap high-and-low order bytes within a PDP-11 word.

2.11 Building the Modified RIPS Task

To construct a new executable task image after modifying one or more of the source modules described previously, it is necessary to perform only three steps:

- a. MCR>SET /UIC = [11, 11]
- b. Recompile or reassemble the affected source modules to obtain corresponding object modules.
- c. MCR>TKB @DK3:RIPSLNK

It is assumed that:

- o RSX-11M system disk is mounted on drive 0
- o MCC:RSX is mounted on drive 3.
- o PAR Development Disk is mounted on drive 2.

This will build the task "RIPS.TSK" on DK3 :[11, 11].

2.12 To Run "RIPS" with the COMTAL

```
MCR>LOA ZB:           (Load the COMTAL Device Driver)
MCR>INS DK3:RIPS       (Install the task)
MCR>RUN RIPS          (Run the task)
```

Once executing, RIPS prompts are self-explanatory.

2.13 Integration of Mc² Software With Other User's Programs

Other users may gain access to the COMTAL under RSX-11M by including certain of the Mc² object modules in their task builder (TKB) command lines. Normally, the modules required will be:

COMLØ1.OBJ
COMLØ2.OBJ
COMLØ3.OBJ
COMLØ4.OBJ
COMLØ5.OBJ
IWRITW.OBJ

This will give the user access to all the I/O functions described in the documentation of COMLØ2 - COMLØ5. If it is determined that all of these functions are not necessary for the user's program, it may be possible to remove one of the modules COMLØ2 - COMLØ5 if none of the functions within the module are required by the user. In any case, COMLØ1 and IWRITW are required, and cannot be omitted.

L1: .RNT DK2:EXEC-ERRFLG-SINACC-L1-L2,L3)
 L4: .FCTR DK2:SETDB-DK2:ERR-DK2:GETNUM-DK2:FSMUT-DK2:FCLOSE-L41
 L5: .FCTR DK2:FCRE-DK2:FCRE-DK2:PRETO-DK2:FILE1-LB1
 L6: .FCTR DK2:BLDISP-DK2:GETEXT-DK2:IFDE-DK2:DSXFR-LC1
 L7: .FCTR DK2:FILE10-DK2:TYI-DK2:YTO-LD1
 L8: .FCTR DK2:TELE10-LF1
 L9: .FCTR DK2:TYGPF-LF1
 L10: .FCTR DK2:PLOT-LG1
 L11: .FCTR DK2:SAVER-LM1
 L12: .FCTR DK2:ADEXTN-DK2:GTEXTN-DK2:DUPCHK-DK2:LOGINP-L11
 L13: .FCTR DK2:SETLUM-LJ1
 L14: .FCTR DK2:TTY10-DK2:CR2DA-DK2:UCLOSE-LK1
 L15: .FCTR DK2:FLTPR-DK2:FPMOU-DK2:LDC-L11
 L16: .FCTR DK2:MOD02-DK2:MODF02-DK2:SARFPS-LM1
 L17: .FCTR DK2:SETCK-DK2:STC-DK2:DBLADD-LM1
 L18: .FCTR DK2:DBLD10-DK2:DBLMUL-DK2:DBLSUB-L01
 L19: .FCTR DK2:FLT-DK2:FLTD-DK2:FLTL-LP1
 L20: .FCTR DK2:FLTD-DK2:ID-DK2:IFX-L01
 L21: .FCTR DK2:LDX-DK2:LEX-DK2:MODD13-LR1
 L22: .FCTR DK2:MOD13-DK2:STRDBL-DK2:STSNGL-L51
 L23: .FCTR DK2:OPTION-DK2:REMAIN
 L24: .FCTR DK2:FSYSIN-DK2:SETSI2-DK2:SETDEF
 L25: .FCTR DK2:CALSUB-(Z1,Z2)
 L26: .FCTR DK2:FPTD-DK2:FPTD-(L4,L5,L6,L7,L8,L9,L10,L11,L12,L14,L15,L33)
 L27: .FCTR DK2:FPTD-DK2:FPTD-(L13,L2)
 L28: .FCTR DK2:TEKDISP
 L29: .FCTR DK2:TAPINP-DK2:TAPSUB
 L30: .FCTR DK2:TAPOUT-DK2:TPREU-DK2:TPSKIP-DK2:TAPSUB
 L31: .FCTR DK2:SINMO-DK2:SMOOTH
 L32: .FCTR DK2:PURGE-DK2:LOGERR-DK2:FRNH-DK2:RENAME
 L33: .FCTR DK2:ELCHNG-DK2:TRMFTN
 L34: .FCTR DK2:USPRG-DK2:FDPTD-DK2:FPTD2
 L35: .FCTR DK2:PTEDGE
 L36: .FCTR DK2:DEWST-DK2:WISPLT-DK2:FPA5C-DK2:FINDPT
 L37: .FCTR DK2:LINES-DK2:SMOOTH2
 L38: .FCTR DK2:NORHLZ
 L39: .FCTR DK2:UTDSNO
 L40: .FCTR DK2:PARCOR-DK2:OUTCAR-DK2:RMFTN-(L17A,L17B,L17C,L17D,L17E)
 L41: .FCTR DK2:ARSLCT
 L42: .FCTR DK2:RTIOMG
 L43: .FCTR DK2:DIFFAC
 L44: .FCTR DK2:COMBIN
 L45: .FCTR DK2:PULINR
 L46: .FCTR DK2:SPBIAS
 L47: .FCTR DK2:SHIFTI
 L48: .FCTR DK2:CKTEXT
 L49: .FCTR DK2:CTFILE
 L50: .FCTR DK2:CTFILE
 L51: .FCTR DK2:ODDL
 L52: .FCTR DK2:DECIM
 L53: .FCTR DK2:MODREP
 L54: .FCTR DK2:SFILL
 L55: .FCTR DK2:CLCU
 L56: .FCTR DK2:DIRCTV
 L57: .FCTR DK2:KEYARA
 L58: .FCTR DK2:LPNDGP-DK2:LNPRINT-DK2:DUMP
 L59: .FCTR

FORTRAN IV U01C-03 MON 29-OCT-79 14:07:09 PAGE 001
 CORE-08X, UIC-E1.123J ,TT0:-DK3:CALSUB

PROGRAM NAME: CALSUB TYPE: SUBROUTINE
 PROGRAMMER: ALBERT J. FRANKLIN DATE OF LAST UPDATE: 15-JAN-79

PROGRAM FUNCTION:
 CALSUB CALLS THE REQUESTED OPTION SUBROUTINE.

ENTRY POINTS AND CALLING SEQUENCE:
 CALL CALSUB (IF,IS,CORE,ISIZE)
 INPUTS: IF,IS,CORE,ISIZE
 OUTPUTS: NONE

DETAILED PROGRAM DESCRIPTION:
 CALSUB GOES TO THE OPTION LIST FOR FRAME IF, AND CHECKS FOR OPTION
 NUMBER IS AMONG THE LIST OF IMPLEMENTED OPTIONS. IF THE OPTION IS
 MATCHED, THE APPROPRIATE SUBROUTINE IS CALLED; IF NOT, AN ERROR
 MESSAGE IS PRINTED AND NO FURTHER ACTION IS TAKEN.

SUBROUTINES CALLED (EXTERNAL):

ABSDIF - ABSOLUTE DIFFERENCE
 ABSUS - UNSTD. ABS. VALUE SMOOTH
 ADPILL - ADAPTIVE FILL
 ARSLOT - RECTANGULAR AREA SELECT
 AVE - AVERAGE IMAGES
 BOXSMO - BOX SMOOTH
 CTEXT - EXAMINE HEADER TEXT
 CLOS - CLOSED CURVE
 CTFIL - CREATE TEST IMAGE
 DECIM - DECIMATION
 DIFFAC - SCALED WEIGHTED COMBINATION
 DIRCTV - DIRECTORY LISTING UTILITY
 EDGPRN - LIST BINARY IMAGE
 ELCHNG - THRESHOLD AND ELEMENT CHANGE
 FDEL - DELETE A DISK FILE
 FRNM - RENAME A DISK FILE
 GRVDMP - LIST IMAGE GREY VALUES
 GTFIL - GENERATE TEST IMAGE
 IMAGE - IMAGE HISTOGRAMS
 KEYVAR - SPECIFY REGION COORDINATES
 LINES - AREA EDGE DETECTION
 LOGERR - CONTROL THE FOUR LOG OPTIONS
 MAED - AREA EDGE DETECTION MAX.
 MODREP - MODAL REPLACEMENT
 NDOT - ODD DOT ELIMINATION
 NORPLZ - NORMALIZE AND RANGE CHANGE
 ODDLIN - ODD LINE ELIMINATION

PTEDGE - POINT EDGE DETECTION
 PULING - ARBITRARY FUNCTION
 REGION - REGION HISTOGRAMS
 RGSTAT - LIST REGION STATISTICS
 RNCING - RANGE CHANGE
 RTIONG - RATIO IMAGES
 SCALE - SCALED DIFFERENCE

FORTRAN IV U01C-03 MON 29-OCT-79 14:07:09 PAGE 002
 CORE=08K, UIC=C1,123J ,TT0:-DK3:CALSUB

SFILL - SIMPLE FILL
 SHIFTI - SHIFT IMAGE
 SINSPO - M X N SMOOTH
 SINGLN - IMAGE ROW INTENSITY PROFILE
 SPIAS - ADD CONSTANT
 TAPING - INPUT AN IMAGE FILE FROM TAPE
 TAPOUT - OUTPUT AN IMAGE FILE TO TAPE
 TEKDISP - DISPLAY A BINARY IMAGE ON THE TEKTRONIX 4014
 THRESH - THRESHOLD IMAGE
 TPEN - REWIND MAGTAPE ON UNIT 0
 TPSKIP - SKIP FILES ON MAGTAPE MOUNTED ON UNIT 0
 TTYOUT - OUTPUT CHARACTER STRING TO TTY
 USRPRG - ANY USER PROGRAM
 WTSPO - WEIGHTED SMOOTH AND ABSOLUTE VALUE WEIGHTED SMOOTH

INTERNAL VARIABLES AND ARRAYS (FORTRAN TYPE CONVENTION):
 IER - ERROR STATUS

GLOBAL VARIABLES AND ARRAYS (FORTRAN TYPE CONVENTION):
 CORE - BYTE ARRAY (ISIZE) USER DATA BUFFER AREA
 IF - FRAME NUMBER
 IS - OPTION NUMBER
 ISIZE - SIZE OF USER DATA BUFFER AREA IN BYTES
 LCOUNT - TEKTRONIX LINE COUNT

SPECIAL COMMENTS:
 THE FREE CORE AREA AND ITS SIZE IN BYTES ARE PASSED THROUGH
 CALSUB TO THE CALL ROUTINES BY PARAMETERS RATHER THAN IN
 COMMON. THIS IS SO ONLY EXEC MUST BE CHANGED WHEN THE SIZE
 OF FREE CORE CHANGES.

SUBROUTINE CALSUB (IF,IS,CORE,ISIZE)

BYTE CORE(ISIZE)

COMMON /LINE/LCOUNT

////////// BEGIN EXECUTABLE INSTRUCTIONS //////////

0001

0002

0003

```

0004      GO TO CODE FOR FRAME IF.
          GO TO (21,12,3,4,5,6,7,21,21,21,11,12,13,14,
1 21,21,21,21,21), IF + 1
          CODE FOR FRAME 1.
          IF (IS .EQ. 1) CALL ABSLCT
          IF (IS .EQ. 2) CALL KEYARA
          IF (IS .EQ. 3) CALL SHIFTI
          IF (IS .EQ. 4) CALL DECIN
          IF (IS .GT. 4) GO TO 21
          RETURN
0005
0007
0009
0011
0013
0015

```

PAGE 003
TTO:DK3:CALSUB

MON 29-OCT-79 14:07:09

FORTRAN IV U01C-03
CORE-08K, UIC-E1,123J

```

0016      CODE FOR FRAME 2.
          IF (IS .EQ. 1) CALL SPBIAS
          IF (IS .EQ. 2) CALL THRSH
          IF (IS .EQ. 3) CALL NORMLZ
          IF (IS .EQ. 4) CALL ELCHNG
          IF (IS .EQ. 5) CALL RNCNG
          IF (IS .EQ. 6) CALL PULINR
          IF (IS .GT. 6) GO TO 21
          RETURN
0018
0020
0022
0024
0026
0028
0030
          CODE FOR FRAME 3
          IF (IS .EQ. 1) CALL IMAGE
          IF (IS .EQ. 2) CALL REGION
          IF (IS .EQ. 3) CALL RGSTAT
          IF (IS .EQ. 5) CALL SNGLM
          IF (IS .EQ. 4 .OR. IS .GT. 5) GO TO 21
          RETURN
0031
0033
0035
0037
0039
0041
          CODE FOR FRAME 4
          IF (IS .EQ. 1) CALL BOXSMO
          IF (IS .EQ. 2) CALL SIRSNO
          IF (IS .EQ. 3) CALL UTSNO
          IF (IS .EQ. 4) CALL ABSUS
          IF (IS .GT. 4) GO TO 21
          RETURN
0042
0044
0046
0048
0050
0052
          CODE FOR FRAME 5
          IF (IS .EQ. 1) CALL AVE
          IF (IS .EQ. 2) CALL SCALE
0053
0055

```

```

0067 IF (IS .EQ. 3) CALL ABSDF
0068 IF (IS .EQ. 4) CALL DIFFAC
0069 IF (IS .EQ. 5) CALL RTIOMG
0070 IF (IS .GT. 5) GO TO 21
0071 RETURN
0072
0073 CODE FOR FRAME 6
0074
0075 IF (IS .EQ. 1) CALL MODREP
0076 IF (IS .EQ. 2) CALL MODT
0077 IF (IS .EQ. 3) CALL ODDLIN
0078 IF (IS .GT. 3) GO TO 21
0079 RETURN
0080
0081 CODE FOR FRAME 7
0082
0083 IF (IS .EQ. 1) CALL PTEDGE
0084 IF (IS .EQ. 2) CALL LINES
0085 IF (IS .EQ. 3) CALL PARED
0086 IF (IS .EQ. 4) CALL SFILL
0087 IF (IS .EQ. 5) CALL ADFILL
0088 IF (IS .EQ. 6) CALL CLCV

```

PAGE 004
TT0:DK3:CALSUB

FORTRAN IV U01C-03 MON 29-OCT-79 14:07:09
CORE=08K, UIC=11,123J

```

0087 IF (IS .EQ. 7) CALL EDGPRN
0088 IF (IS .GT. 7) GO TO 21
0089 RETURN
0090
0091 CODE FOR FRAME 11.
0092
0093 IF (IS .EQ. 1) CALL FDEL (-1,IER)
0094 IF (IS .EQ. 2) CALL FRM (-1,IER)
0095 IF (IS .EQ. 3) OR IS .EQ. 4) CALL DIRCTY
0096 IF (IS .EQ. 5) CALL TAPINP (CORE,ISIZE,IER)
0097 IF (IS .EQ. 6) CALL TAPOUT (CORE,ISIZE,IER)
0098 IF (IS .EQ. 7) CALL TPKIP (IER)
0099 IF (IS .EQ. 8) CALL TPREV (IER)
0100 IF (IS .LE. 8) RETURN
0101 GO TO 21
0102
0103 CODE FOR FRAME 12.
0104
0105 IF (IS .GT. 1) GO TO 21
0106 IF (IS .EQ. 1) CALL TEKDSP
0107 RETURN
0108
0109 CODE FOR FRAME 13
0110
0111 IF (IS .EQ. 2) CALL LOGERR (IER)

```

```

0110 IF (IS .EQ. 3) CALL USMRG
0111 IF (IS .EQ. 4) CALL CTFILF
0112 IF (IS .EQ. 5) CALL CTFILF
0120 IF (IS .EQ. 6) CALL CKTEXT
0121 IF (IS .EQ. 7) CALL GRVDMP
0126 IF (IS .EQ. 1 .OR. IS .GT. 7) GO TO 21
0128 RETURN
      CODE FOR FRAME 14
0129 IF (IS .EQ. 1) CALL IUIMF
0131 IF (IS .EQ. 2) CALL IURMF
0133 IF (IS .EQ. 3) CALL IUGMF
0135 IF (IS .EQ. 4) CALL IURMF
0137 IF (IS .EQ. 5) CALL IURMF
0139 IF (IS .EQ. 6) CALL IURMF
0141 IF (IS .EQ. 7) CALL IURMF
0143 IF (IS .EQ. 8) CALL IURMF
0145 IF (IS .EQ. 9) CALL IURMF
0147 IF (IS .EQ. 0 .OR. IS .GT. 9) GO TO 21
0149 RETURN
      PRINT MESSAGE SAYING SUBROUTINE NOT OPERATIVE.
      CALL TTYOUT ('NOT OPERATIVE')
      LCOUNT = LCOUNT + 1
      RETURN
      END

```

FORTRAN IU		STORAGE MAP	
NAME	OFFSET	ATTRIBUTES	
CORE	000020	LOGICALX1	PARAMETER ARRAY (ISIZE)
ISIZE	000022	INTEGERX2	PARAMETER VARIABLE
IF	000014	INTEGERX2	PARAMETER VARIABLE
IS	000016	INTEGERX2	PARAMETER VARIABLE
ARSLCT	000000	REALX4	PROCEDURE
KEYAPA	000000	INTEGERX2	PROCEDURE
SHIFTI	000000	REALX4	PROCEDURE
DECIM	000000	REALX4	PROCEDURE
SPRIAS	000000	REALX4	PROCEDURE
THRSH	000000	REALX4	PROCEDURE
NORMLZ	000000	INTEGERX2	PROCEDURE
ELCHNG	000000	REALX4	PROCEDURE
PULINR	000000	REALX4	PROCEDURE
IMAGE	000000	INTEGERX2	PROCEDURE
REGION	000000	REALX4	PROCEDURE
RGSTAT	000000	REALX4	PROCEDURE
SNGLH	000000	REALX4	PROCEDURE
BOXSNO	000000	REALX4	PROCEDURE
SIMSNO	000000	REALX4	PROCEDURE
UTSNO	000000	REALX4	PROCEDURE


```

FORTRAN IU          U01C-03          PAGE 001
CORE-08X, UIC=E1,1233          ,TT0:DX31CIPSB

MON 29-OCT-79 14:08:35          FILE NAME: CIPSB.FTN
                                DATE LAST EDITED: 21-JUN-79
                                10-JUL-79

HISTORY:
DATE:
-----
19-MAR-79      INITIAL WRITING
21-MAR-79      ADD MORE SUBS
18-APR-79      ADD CLEAR SCREEN
                TO 'IUIF', AND
                ADD 'IUCBF'
21-JUN-79      CONVERSION TO RIPS

                                THIS FILE CONTAINS THE MAIN SUBROUTINES FOR THE
                                RIPS CORTAL MANIPULATIONS AND CALLS SEVERAL OF THE CORTAL LIBRARY
                                ROUTINES AS WELL AS OTHER RIPS FILE READING ROUTINES.

                                THE MAIN ROUTINES ARE:

                                IUIF - READ A RIPS FILE DISPLAY ON CORTAL
                                IRIW - WRITE A RIPS FILE FROM THE CORTAL
                                IUCW - READ A RIPS GRAPHICS FILE TO CORTAL
                                IRCW - WRITE A RIPS GRAPHICS FILE FROM CORTAL
                                IURF - READ A RIPS FUNCTION FILE TO CORTAL
                                IIRF - WRITE A RIPS FUNCTION FILE FROM CORTAL
                                IUPSF - READ A RIPS PSEUDO-COLOR FILE TO CORTAL
                                IIRPSF - WRITE A RIPS PSEUDO-COLOR FILE FROM CORTAL
                                IUCBF - WRITE COMMAND BLOCK TO CORTAL

SUBROUTINE IUIF
      FUNCTIONAL DESCRIPTION:
      THIS ROUTINE WILL READ A RIPS ' IMAGE ' TYPE FILE AND
      WRITE IT TO ONE OF THE IMAGE MEMORIES IN THE CORTAL.

      1) QUERY USER FOR IMAGE NUMBER

```



```

C
0032 C DO 100 K=1,NRECS
0032 C J = K -1 + IVOFF
C
0034 CALL FPTR(LUN,FRECR(IXOFF),256,K,1,0,LPTR,IEOF,IER)
0035 IEP=IBYTEX(FRECR,256)
0036 IEP=IUM(IMNO,J,FRECR)
C
0037 100 CONTINUE
0038 C CALL FCLOSE(LUN,IER)
C

```

FORTRAN IV U01C-03 MON 29-OCT-79 14:08:35 PAGE 003
 CORE=08K, UIC=E1,123 ,TT0=DK3:CP1P58

```

0039 RETURN
0040 END

```

NAME	OFFSET	STORAGE MAP	ATTRIBUTES
ITEAT	000014	LOGICALX1	ARRAY (8)
IM5	000024	LOGICALX1	ARRAY (4)
IMTYP	000030	INTEGERX2	ARRAY (2)
ONE	000034	LOGICALX1	VARIABLE
TWO	000035	LOGICALX1	VARIABLE
THREE	000036	LOGICALX1	VARIABLE
FOUR	000037	LOGICALX1	VARIABLE
ICNT	000046	INTEGERX2	VARIABLE
SHGDEC	000000	REALX4	PROCEDURE
IMNO	000050	INTEGERX2	VARIABLE
IEP	000052	INTEGERX2	VARIABLE
FRET0	000000	REALX4	PROCEDURE
NRECS	000054	INTEGERX2	VARIABLE
NRECSZ	000056	INTEGERX2	VARIABLE
LUN	000060	INTEGERX2	VARIABLE
IXOFF	000062	INTEGERX2	VARIABLE
IYOFF	000064	INTEGERX2	VARIABLE
J	000066	INTEGERX2	VARIABLE
IMCB	000000	INTEGERX2	PROCEDURE

```
COMMON BLOCK /FRECOR/  LENGTH 020000
FREECOR 000000 INTEGER*2 ARRAY (4096)
COMMON BLOCK /IMAGE/  LENGTH 000002
IMAGE 000000 INTEGER*2 VARIABLE
COMMON BLOCK /INNAM/  LENGTH 000062
INNAM 000000 LOGICAL*1 ARRAY (50)
COMMON BLOCK /NANTAB/  LENGTH 000040
NAME 000000 LOGICAL*1 ARRAY (32)
COMMON BLOCK /CONTAL/  LENGTH 001000
BUF 000000 INTEGER*2 ARRAY (256)
```

```

SUBROUTINE IRIMF
,
, FUNCTIONAL DESCRIPTION:
,
, THIS ROUTINE WILL READ A COMTAL IMAGE AND
, WRITE IT TO A RIPS ' IMAGE ' FILE.
,
,
, 1) QUERY USER FOR IMAGE NUMBER
, 2) OPEN FILE FOR READ/WRITE
, 3) READ COMTAL, WRITE FILE
,
,
, INTEGER NAFTYP(2),FREECR(4096)
, LOGICAL L1_NAME(32)
, INTEGER BUF(256)
,
, COMMON /OUTNAM/,ONAM/,COMTAL/BUF/,FREECR/,FREECR
, COMMON /INATAB/,NAME/,IMAGE/,IMAGE

```

```

0007 DATA NAFTYP /'.I.', 'PG'//
      C
0008 1) QUERY USER
0009 ICNT=1
      C CALL SNGDEC(LIMNO, ICNT, IMNO, IER)
      C READ CORTAL, WRITE RIPS FILE
      C
0010 CREATE THE RIPS FILE BEFORE WRITING
      C CALL FCRCO(ONAM, 256, 256, NAME, NAFTYP, LUN, BUF, IER)
      C
0011 DO 200 J=1, 256
0012 IER=IRIM(LIMNO, J, FREECR)
0013 IER=IBYTEX(FREECR, 256)
      C READ THE IMAGE FROM THE CORTAL
      C
0014 CALL FPTR(LUN, FREECR, 256, J, 1, 1, LPTR, IEOP, IER)
0015 NOW WRITE IT TO THE FILE
      C CONTINUE
0016 CALL FCLOSE(LUN, IER)
      C
0017 RETURN
0018 END

```

FORTRAN IV		STORAGE MAP	
NAME	OFFSET	ATTRIBUTES	
NAFTYP	000014	INTEGER12 ARRAY (2)	
ICNT	000024	INTEGER12 VARIABLE	
SNGDEC	000030	REAL14 PROCEDURE	
IMNO	000036	INTEGER12 VARIABLE	
IER	000039	INTEGER12 VARIABLE	
FCRCO	000040	REAL14 PROCEDURE	
LUN	000032	INTEGER12 VARIABLE	
J	000034	INTEGER12 VARIABLE	
IRIM	000040	INTEGER12 PROCEDURE	
IBYTEX	000040	INTEGER12 PROCEDURE	
FPTR	000040	REAL14 PROCEDURE	
LPTR	000036	INTEGER12 VARIABLE	
IEOP	000040	INTEGER12 VARIABLE	
FCLOSE	000040	REAL14 PROCEDURE	
COMMON BLOCK /OUTNAM/		LENGTH 000004	
ONAM	000000	REAL14 VARIABLE	
COMMON BLOCK /CORTAL/		LENGTH 001000	

```

BUF      000000 INTEGER12 ARRAY (256)
COMMON BLOCK /FRECOR/  LENGTH 020000
FREECR  000000 INTEGER12 ARRAY (4096)
COMMON BLOCK /NARTAB/  LENGTH 000040
NAME     000000 LOGICAL11 ARRAY (32)
COMMON BLOCK /IMAGE/   LENGTH 000002
IMAGE    000000 INTEGER12 VARIABLE

```

FORTRAN IV U01C-03 MON 29-OCT-79 14:09:31 PAGE 001
 CORE=08K, UIC=C1,123J ,TT0=-DK3:CP1PSB

```

0001      SUBROUTINE IUPSF
          , FUNCTIONAL DESCRIPTION:
          , THIS ROUTINE WILL READ A RIPS PSEUDO-COLOR FILE
          , AND WRITE IT TO THE CORTAL PSEUDO-COLOR MEMORY.
          ,
          , 1) OPEN FILE FOR READ/WRITE
          , 2) READ FILE, WRITE CORTAL
          , 3) READ FILE, WRITE CORTAL
          , 4) , CLOSE RIPS FILE
          ,
          INTEGER BUF(256),PSE(2),FREECR(4096)
          LOGICAL11 NAME(12)
          COMMON /INNAM/INAM2,/FRECOR/FRECR,/CORTAL/BUF
          COMMON /NARTAB/NAME
          DATA PSE /'.P','SE'/
          ,
          CALL FRET0(INNAM,NRECS,NRECS2,NAME,PSE,LUN,BUF,IER)
          ,
          , 4) READ RIPS FILE, WRITE TO CORTAL PSEUDO-COLOR
          CALL FPTR(LUN,FRECR,512,1,2,0,LPTR,IEOF,IER)
          IER = IUPS (FREECR)
          , 5) CLOSE RIPS FILE, CLOSE CORTAL
          ,
          CALL
          RETURN
          END
0010
0011
0012

```

FORTRAN IV STORAGE MAP

NAME	OFFSET	ATTRIBUTES
PSE	000014	INTEGER*2 ARRAY (2)
PRETO	000000	REAL*4 PROCEDURE
NRECS	000030	INTEGER*2 VARIABLE
NRECS2	000032	INTEGER*2 VARIABLE
LUN	000034	INTEGER*2 VARIABLE
IER	000036	INTEGER*2 VARIABLE
EDTR	000000	REAL*4 PROCEDURE
LPTR	000040	INTEGER*2 VARIABLE
IEOF	000042	INTEGER*2 VARIABLE
IUPS	000000	INTEGER*2 PROCEDURE
FCLOSE	000000	REAL*4 PROCEDURE
COMMON BLOCK /IMNAM/ LENGTH 000002		
INAM	000000	INTEGER*2 VARIABLE
COMMON BLOCK /FRECOR/ LENGTH 020000		
FRECR	000000	INTEGER*2 ARRAY (4096)
COMMON BLOCK /CORTAL/ LENGTH 001000		
BUF	000000	INTEGER*2 ARRAY (256)
COMMON BLOCK /NAMTAB/ LENGTH 000040		
NAME	000000	LOGICAL*1 ARRAY (32)

FORTRAN IV U01C-03 MON 29-OCT-79 14:09:42 PAGE 001
 CORE=08K, UIC=C1,123 ,TT0:-DK3:CP1P58

0001 C SUBROUTINE IRPSF
 C , FUNCTIONAL DESCRIPTION:
 C ; THIS ROUTINE WILL READ THE CORTAL PSEUDO-COLOR MEMORY
 C ; AND WRITE IT TO A RIPS PSEUDO-COLOR FILE.

```

00000000
0002      1) OPEN FILE FOR READ/WRITE
0003      2) READ CORTAL, WRITE FILE
0004      3) CLOSE RIPS FILE
0005      4) READ CORTAL, WRITE TO RIPS PSEUDO-COLOR
0006      5) CLOSE RIPS FILE, CLOSE CORTAL
0007      CALL FCLOSE(LUN,IER)
0008      RETURN
0009      END
0010      COMMON /FRECOR/ FRECOR
0011      COMMON /OUTNAM/ ONAM, /CORTAL/ BUF, /NARTAB/ NAME
0012      DATA NAFTYP /'P', SE, /
0013      1)
0014      NRECS = 1
0015      NRECSZ = 256
0016      CALL FCREC(ONAM,NRECSZ,2,NAME,NAFTYP,LUN,BUF,IER)
0017      4) READ CORTAL, WRITE TO RIPS PSEUDO-COLOR
0018      IER = IRPS (FRECOR)
0019      CALL FPTR(LUN,FRECOR,NRECSZ,1,2,1,LPTR,IEOF,IER)
0020      5) CLOSE RIPS FILE, CLOSE CORTAL
0021      CALL FCLOSE(LUN,IER)
0022      RETURN
0023      END
0024      COMMON BLOCK /FRECOR/ LENGTH 820000
0025      FRECOR 000000 INTEGER32 ARRAY (4000)

```

```

FORTRAN IV      STORAGE MAP
NAME  OFFSET  ATTRIBUTES
NAFTYP 000014  INTEGER32  ARRAY (2)
NRECS  000024  INTEGER32  VARIABLE
NRECSZ 000026  INTEGER32  VARIABLE
FRECOR 000030  REAL32    PROCEDURE
LUN     000032  INTEGER32  VARIABLE
IER     000034  INTEGER32  PROCEDURE
IRPS    000036  REAL32    PROCEDURE
FPTR    000038  INTEGER32  VARIABLE
LPTR    000040  INTEGER32  VARIABLE
IEOF    000042  REAL32    PROCEDURE
FCLOSE  000044  REAL32    PROCEDURE
COMMON BLOCK /FRECOR/ LENGTH 820000
FRECOR 000000 INTEGER32 ARRAY (4000)

```

```

COMMON BLOCK OUTNAM/ LENGTH 000004
COMMON 000000 REAL14 VARIABLE
COMMON BLOCK /CONTAL/ LENGTH 001000
BUF 000000 INTEGER12 ARRAY (256)
COMMON BLOCK /NANTAB/ LENGTH 000040
NAME 000000 LOGICAL11 ARRAY (32)

```

FORTRAN IV U01C-03 MON 29-OCT-79 14:10:04 PAGE 001
 CORE-08K, UIC-11,123 ,TTO-DK3ICPISB

```

0001 C SUBROUTINE IUGHF
0002 C
0003 C FUNCTIONAL DESCRIPTION:
0004 C
0005 C THIS ROUTINE WILL READ A RIPS GRAPHICS (.GRA) FILE
0006 C AND WRITE IT TO THE CONTAL GRAPHICS MEMORY.
0007 C
0008 C 1) OPEN FILE FOR READ/WRITE
0009 C 2) READ FILE, WRITE CONTAL
0010 C 3) CLOSE RIPS FILE
0011 C
0012 C INTEGER BUF(256), FREECR(4096), GRA(2)
0013 C LOGICAL11 NAME(32)
0014 C COMMON /FREECR/ FREECR, /CONTAL/ BUF, /NANTAB/ NAME
0015 C COMMON /INNAM/ INNAM, /IGRAQU/ IGRAQU
0016 C DATA GRA /'.G.', RA /
0017 C
0018 C 1) QUERY USER FOR GRAPHICS NUMBER
0019 C
0020 C ICNT=1
0021 C CALL SNGDEC(IGRAQU, ICNT, INNO, IER)
0022 C
0023 C 2) OPEN RIPS FILE
0024 C CALL FRET0(INNAM, NRECS, NRECSZ, NAME, GRA, LUN, BUF, IER)
0025 C
0026 C 4) READ RIPS FILE, WRITE TO CONTAL GRAPHICS MEMORY
0027 C
0028 C DO 100 J=1,256
0029 C
0030 C CALL FPTR(LUN, FREECR, 256, J, 1, 0, LPTR, IE0F, IER)
0031 C IER = IUGH (INNO, J, FREECR)
0032 C
0033 C 100 CONTINUE

```



```

      5) CLOSE RIPS FILE, CLOSE CORTAL
      CALL FCLOSE(LUN,IER)
      RETURN
      END
0014
0015
0016

```

```

FORTRAN IU      STORAGE MAP
NAME  OFFSET  ATTRIBUTES
GRA  000014  INTEGER32 ARRAY (2)
ICNT 000026  INTEGER32 VARIABLE
SNGDEC 000030  REAL34 PROCEDURE
IMNU 000032  INTEGER32 VARIABLE
IER 000034  INTEGER32 VARIABLE
FRET0 000036  REAL34 PROCEDURE
MRECS 000038  INTEGER32 VARIABLE
MRECSZ 000040  INTEGER32 VARIABLE
LUN 000042  INTEGER32 VARIABLE
J 000044  REAL34 PROCEDURE
FPTR 000046  INTEGER32 VARIABLE
LPTR 000048  INTEGER32 VARIABLE
IEOF 000050  INTEGER32 VARIABLE
IUCM 000052  INTEGER32 PROCEDURE
FCLOSE 000054  REAL34 PROCEDURE

COMMON BLOCK /FREC0/  LENGTH 020000
FREC0 000000  INTEGER32 ARRAY (4096)

COMMON BLOCK /CORTAL/  LENGTH 001000
BUF 000000  INTEGER32 ARRAY (256)

COMMON BLOCK /MARTAB/  LENGTH 000040
NAME 000000  LOGICAL31 ARRAY (32)

COMMON BLOCK /INMAN/  LENGTH 000002
INMAN 000000  INTEGER32 VARIABLE

COMMON BLOCK /IGRAOU/  LENGTH 000002
IGRAOU 000000  INTEGER32 VARIABLE

```

FORTMAN IV U01C-03 MON 29-OCT-79 14:10:17 PAGE 001
 CORE-08X, UIC-C1.123J ,TTO:-DK3:CP1PSB

```

0001 SUBROUTINE IRGWF
C
C      FUNCTIONAL DESCRIPTION:
C      THIS ROUTINE WILL READ THE CORTAL GRAPHICS MEMORY
C      AND WRITE IT TO A RIPS GRAPHICS (.GRA) FILE.
C
C      1) QUERY USER FOR GRAPHICS NUMBER
C      2) OPEN FILE FOR READ/WRITE
C      3) READ CORTAL WRITE FILE
C      5) , CLOSE RIPS FILE
C
C      INTEGER BUF(256),FREECR(4096),NAFTYP(2)
C      LOGICAL L1, NAME(32)
C      COMMON /OUTNAM/ONAM,/FREC0R/FRECSZ,/NAFTAB/NAME
C      COMMON /CORTAL/BUF,/IGRAQU/IGRAQU
C      DATA NAFTYP /'.G','RA'/
C
C      1) QUERY USER FOR GRAPHICS NUMBER
C      ICHT=1
C      CALL SNGDEC(IGRAQU,ICHT,IMNO,IER)
C      NRECS= 256
C      NRECSZ= 32
C      CALL FOREO(ONAM,NRECS,NRECSZ,NAME,NAFTYP,LUN,BUF,IER)
C      4) READ CORTAL, WRITE TO RIPS GRAPHICS
C      DO 100 J=1,256
C      IER = IRGM (IMNO,J,FRECR)
C      CALL FPTR(LUN,FRECR,256,J,1,1,LPTR,IEOF,IER)
C      CONTINUE
C      5) CLOSE RIPS FILE.
C      CALL FCLOSE(LUN,IER)
C      RETURN
C      END
0002
0003
0004
0005
0006
0007
0008
0009
0010
0011
0012
0013
0014
0015
0016
0017
0018

```

FORTRAN IV	STORAGE MAP
NAME	OFFSET ATTRIBUTES
NAME	000014 INTEGER12 ARRAY (2)
ICNT	000024 INTEGER12 VARIABLE
SNGDEC	000030 REAL14 PROCEDURE
IMNO	000026 INTEGER12 VARIABLE
IER	000030 INTEGER12 VARIABLE
NRECS	000032 INTEGER12 VARIABLE
NRECSZ	000034 INTEGER12 VARIABLE
FCREC	000036 REAL14 PROCEDURE
LUN	000036 INTEGER12 VARIABLE
J	000040 INTEGER12 VARIABLE
IRGM	000040 INTEGER12 PROCEDURE
FPTR	000040 REAL14 PROCEDURE
LPTR	000042 INTEGER12 VARIABLE
IEOF	000044 INTEGER12 VARIABLE
FCLOSE	000046 REAL14 PROCEDURE
COMMON BLOCK /OUTNAM/ LENGTH 000004	
ONAM	000000 REAL14 VARIABLE
COMMON BLOCK /FRECOR/ LENGTH 020000	
FRECOR	000000 INTEGER12 ARRAY (4096)
COMMON BLOCK /NARTAB/ LENGTH 000040	
NAME	000000 LOGICAL11 ARRAY (32)
COMMON BLOCK /CORTAL/ LENGTH 001000	
EUF	000000 INTEGER12 ARRAY (256)
COMMON BLOCK /IGRAOU/ LENGTH 000002	
IGRAOU	000000 INTEGER12 VARIABLE

```

      FUNCTIONAL DESCRIPTION:
      THIS ROUTINE WILL READ A RIPS FUNCTION (.TRA ) FILE
      AND WRITE IT TO THE CORTAL FUNCTION MEMORY.

      1) QUERY USER FOR FUNCTION NUMBER
      2) OPEN FILE FOR READ/WRITE
      3) READ FILE, WRITE CORTAL
      5) CLOSE RIPS FILE

      LOGICALS: NAME(32)
      INTEGER BUF(256), TRA(2)
      COMMON /FREC/ FRECR, /INNAM/ INAM, /CORTAL/ BUF
      COMMON /NANTAB/ NAME, /IFUNQU/ IFUNQU
      DATA TRA /'.T.', 'RA'

      1) QUERY USER FOR FUNCTION NUMBER
      ICNT=1
      CALL SNGDEC(IFUNQU, ICNT, INNO, IER)
      CALL FRETO(INAM, NRECS, NRECSZ, NAME, TRA, LUN, BUF, IER)
      4) READ RIPS FILE, WRITE TO CORTAL FUNCTION MEMORY
      CALL FPTR(LUN, FRECR, SIZ, 1, 2, 0, LPTR, IEOF, IER)
      IER = IUFM (INNO, FRECR)
      5) CLOSE RIPS FILE,

      CALL FCLOSE(LUN, IER)
      RETURN
      END

```

FORTPAN IU		STORAGE MAP	
NAME	OFFSET	ATTRIBUTES	
TRA	000014	INTEGER12	ARRAY (2)
ICNT	000030	INTEGER12	VARIABLE
SNGDEC	000000	REAL14	PROCEDURE
INNO	000032	INTEGER12	VARIABLE
IER	000034	INTEGER12	VARIABLE
FRETO	000000	REAL14	PROCEDURE
NRECS	000036	INTEGER12	VARIABLE
NRECSZ	000040	INTEGER12	VARIABLE
LUN	000042	INTEGER12	VARIABLE
FPTR	000000	REAL14	PROCEDURE
LPTR	000044	INTEGER12	VARIABLE

```

      IENF 000046 INTEGER12 VARIABLE
      IUPH 000000 INTEGER12 PROCEDURE
      FOLSE 000000 REAL14
      COMMON BLOCK /FRECOR/ LENGTH 000004
      FRECR 000000 REAL14 VARIABLE
      COMMON BLOCK /INNAM/ LENGTH 000002
      INAM 000000 INTEGER12 VARIABLE
      COMMON BLOCK /CORTAL/ LENGTH 001000
      BUF 000000 INTEGER12 ARRAY (256)
      COMMON BLOCK /NARTAB/ LENGTH 000040
      NAME 000000 LOGICAL11 ARRAY (32)
      COMMON BLOCK /IFUNQU/ LENGTH 000002
      IFUNQU 000000 INTEGER12 VARIABLE

```

```

      FORTRAN IU      U01C-03      MON 29-OCT-79 14:11:00      PAGE 001
      CORE-08K, UIC-E1.123J      ,TT0:=DK3:CPIPSB

```

```

0001 C SUBROUTINE IRFHF
      C
      C FUNCTIONAL DESCRIPTION:
      C
      C THIS ROUTINE WILL READ THE CORTAL FUNCTION MEMORY
      C AND WRITE IT TO A RIPS TRANSFER FUNCTION (.TRA) FILE.
      C
      C 1) QUERY USER FOR FUNCTION NUMBER
      C 2) OPEN FILE FOR READ/WRITE
      C 3) READ CORTAL; WRITE FILE
      C 4) . CLOSE RIPS FILE
      C
      C INTEGER TRA(2),FRECR(4096),BUF(256)
      C LOGICAL11 NAME(32)
      C COMMON /OUTNAM/ONAM,/CORTAL/BUF,/FRECOR/FRECR,/NARTAB/NAME
      C COMMON /IFUNQU/IFUNQU
      C DATA TRA /'.T.', 'RA' /
      C
      C 1) QUERY USER FOR FUNCTION NUMBER
      C ICNT=1
      C CALL
      C SNGDEC(IFUNQU,ICNT,IRNO,IER)
      C NRECS = 2

```

```

0010      NRECSZ= 256
0011      CALL FOREIO(ONAM,NRECS,NRECSZ,NAME,TRA,LUM,BUF,IER)
0012      3) READ CONTAL, WRITE TO RIPS TRANSFER FUNCTION FILE
      IER = IFRN (IMNO,FRECR)
0013      CALL FPTR(LUM,FRECR,S12.1.2.1.1.LPTR,IEOF,IER)
      5) CLOSE RIPS FILE,
0014      CALL FCLOSE(LUM,IER)
0015      RETURN
0016      END

```

FORTRAN IV STORAGE MAP

NAME	OFFSET	ATTRIBUTES
TRA	000014	INTEGER12 ARRAY (2)
ICNT	000026	INTEGER12 VARIABLE
SNGDEC	000000	REAL14 PROCEDURE
IMNO	000030	INTEGER12 VARIABLE
IER	000032	INTEGER12 VARIABLE
NRECS	000034	INTEGER12 VARIABLE
NRECSZ	000036	INTEGER12 VARIABLE
FCREO	000000	REAL14 PROCEDURE
LUM	000040	INTEGER12 VARIABLE
IPFN	000000	INTEGER12 PROCEDURE
FPTR	000000	REAL14 PROCEDURE
LPTR	000042	INTEGER12 VARIABLE
IEOF	000044	INTEGER12 VARIABLE
FCLOSE	000000	REAL14 PROCEDURE
COMMON BLOCK /OUTNAM/ LENGTH 000004		
ONAM	000000	REAL14 VARIABLE
COMMON BLOCK /CONTAL/ LENGTH 001000		
BUF	000000	INTEGER12 ARRAY (256)
COMMON BLOCK /FRECOR/ LENGTH 020000		
FRECR	000000	INTEGER12 ARRAY (4096)
COMMON BLOCK /NANTAB/ LENGTH 000040		
NAME	000000	LOGICAL11 ARRAY (32)
COMMON BLOCK /IFUNQU/ LENGTH 000002		

IFUNU 00000 INTEGER12 VARIABLE

FORTRAN IV U01C-03 MON 29-OCT-79 14:11:21 PAGE 001
CORE-08L, UIC-C1.123J ,TT01-DK31CPIPSB

```

0001 C SUBROUTINE IUCBF
      C
      C THIS ROUTINE WILL SEND ONE LINE OF TEXT
      C TO THE COMIAL COMMAND BUFFER
      C
      C 1) QUERY USER FOR TEXT
      C 3) SEND TO COMIAL
      C
      C INTEGER ISTART(1), ITTBUF(40)
      C EQUIVALENCE (ITTBUF(2), ISTART)
      C COMMON /ICOMQU/ICOMQU, ITTBUF/NSIZE, ITTBUF
      C
      C 1) QUERY USER
      C CALL
      C TTOUT( ICOMQU )
      C CALL
      C TTIN ( ITTBUF )
      C 3) SEND TO COMIAL
      C IER= IUCB( ISTART )
      C
      C THIS FUNCTION MUST START ONN THE SECOND WORD OF THE INPUT BUFFER
      C FOR REASONS NOT ENTIRELY CLEAR. IT IS AN IMPLICIT FIX. THE
      C BUFFER CONTAINS LEFT OVER INFORMATION WHICH COMES FROM THE
      C RIPS MENU PROCESSOR. NO WAY TO DATE HAS BEEN DISCOVERED TO
      C REMOVE THE GARBAGE, SO WE JUST START 1 WORD PAST IT.
      C RETURN
      C END
0002
0003
0004
0005
0006
0007
0008
0009

```

FORTRAN IV		STORAGE MAP	
NAME	OFFSET	ATTRIBUTES	
TTOUT	000000	REAL14	PROCEDURE
TTIN	000000	REAL14	PROCEDURE
IER	000014	INTEGER12	VARIABLE
IUCB	000000	INTEGER12	PROCEDURE
COMMON BLOCK /ICOMQU/			LENGTH 000002

```

100000 000000 INTEGER2 VARIABLE
COMMON BLOCK /TTVBUF/ LENGTH 000122
MSIZE 000000 INTEGER2 VARIABLE
ITRBUF 000002 INTEGER2 ARRAY (40)
ISTART 000004 INTEGER2 ARRAY (1)
FOR:

```


FILE NAME: COML01.FOR

DATE LAST EDITED: 8-MAR-79

HISTORY:

DATE:	MODIFICATION
28-FEB-79	INITIAL WRITING
8-MAR-79	CHANGE FUNCTION
	NAMES TO START WITH I
18-SEP-79	CONVERSION TO RIPS

CONTAL LIBRARY FUNCTIONS

THIS CONTAL LIBRARY CONSISTS OF A SERIES OF FUNCTION ROUTINES
 THE FUNCTIONS PROVIDE BASIC BUILDING BLOCKS FOR USER PROGRAMS
 TO USE THE CONTAL.

THE SOURCE FOR THESE FUNCTIONS ARE IN 5 FILES. THEY ARE:

FILE NAME	FUNCTIONS
COML01.FOR	DATA AND BIT PATTERNS FOR CONTAL
COML02.FOR	TRANSFER CODE 1 TYPE
COML03.FOR	TRANSFER CODE 2 TYPE
COML04.FOR	TARGET POSITION
COML05.FOR	TRANSFER CODE 3 TYPE

THE FOLLOWING FUNCTIONS ARE PROVIDED:

N/A	DATA FOR CONTAL COMMUNICATION
1	IUTN WRITE IMAGE MEMORY
2	IRIN NOT DEFINED
3	IUCH READ IMAGE MEMORY
4	IRIN WRITE GRAPHIC MEMORY
5	IUCH NOT DEFINED
6	IRIN READ GRAPHIC MEMORY
7	IUPS WRITE PSEUDO MEMORY
8	IRPS READ PSEUDO MEMORY
9	IUPH WRITE FUNCTION MEMORY
10	IRPH READ FUNCTION MEMORY
11	IUTP WRITE TARGET POSITION

~~~~~

FORTAN IV U01C-03  
CORE-08K, UIC-[1,123]

66

...

1

•

..

\*\*\*

1000

0002  
0003  
0004  
0005

```

C
0005 INTEGER DBB(8,20)
0006 INTEGER TCC
C
0008 INTEGER DB1(8),DB2(8),DB3(8),DB4(8),DB5(8)
0009 INTEGER DB6(8),DB7(8),DB8(8),DB9(8),DB10(8)
0010 INTEGER DB11(8),DB12(8),DB13(8),DB14(8),DB15(8)
0011 INTEGER DB16(8),DB17(8),DB18(8),DB19(8),DB20(8)
C
0012 LOGICAL L1 SPACE
C
0013 EQUIVALENCE (DB1 ,DBB(1,1 ))
0014 EQUIVALENCE (DB2 ,DBB(1,2 ))

```

```

FORTRAN IV      V01C-03      MON 29-OCT-79 14:12:16      PAGE 003
CORE-08K, UIO-C1.1233      ,TT01-DK3:COML01.FOR

```

```

0015 EQUIVALENCE (DB3 ,DBB(1,3 ))
0016 EQUIVALENCE (DB4 ,DBB(1,4 ))
0017 EQUIVALENCE (DB5 ,DBB(1,5 ))
0018 EQUIVALENCE (DB6 ,DBB(1,6 ))
0019 EQUIVALENCE (DB7 ,DBB(1,7 ))
0020 EQUIVALENCE (DB8 ,DBB(1,8 ))
0021 EQUIVALENCE (DB9 ,DBB(1,9 ))
0022 EQUIVALENCE (DB10,DBB(1,10))
0023 EQUIVALENCE (DB11,DBB(1,11))
0024 EQUIVALENCE (DB12,DBB(1,12))
0025 EQUIVALENCE (DB13,DBB(1,13))
0026 EQUIVALENCE (DB14,DBB(1,14))
0027 EQUIVALENCE (DB15,DBB(1,15))
0028 EQUIVALENCE (DB16,DBB(1,16))
0029 EQUIVALENCE (DB17,DBB(1,17))
0030 EQUIVALENCE (DB18,DBB(1,18))
0031 EQUIVALENCE (DB19,DBB(1,19))
0032 EQUIVALENCE (DB20,DBB(1,20))

```

```

C
0033 COMMON /DBB/ DBB
0034 COMMON /TC/ TCUIM, TCRIM, TCUCH, TCRCH, TCUPS, TCRPS,
*          TCUFH, TCRFH, TCUFP, TCRFP, TCUDC, TCRDC,
*          TCUCB, TCRCB, TCUCD, TCRCD, TCUDI, TCRDI
C
0035 COMMON /CONVAR/ MAXIM,MAXLN,MAXGM
0036 COMMON /DATAA/ TCC

```

```

* THE FOLLOWING DEFINE THE NUMBER OF CORTAL IMAGES, NUMBER OF
* LINES IN AN IMAGE, AND THE NUMBER OF GRAPHICS,
* NOTE: ALL NUMBERS ARE 1 LESS ACTUAL

```

```

0037      (0 - 2, IS 3 MEMORIES)
          DATA MAXIM /2/, MAXLN / 255 /, MAXGM / 2 /

          ;
          ; THE FOLLOWING TABLE DEFINES THE DBS FOR THE
          ; VARIOUS FUNCTIONS DB1 IS USED FOR FUNCTION 1
          ; WHICH IS WRITE IMAGE MEMORY (WIM) THIS IS
          ; DETERMINED BY TCUM-1.
          ;
          DATA DB1 / 101000 /, 0 /, 0 /, 0 /, -128 /, 2403 /, 3 /, 0 /
          DATA DB2 / 101000 /, 0 /, 0 /, 0 /, -64 /, 1403 /, 3 /, 0 /
          DATA DB3 / 10000 /, 10000 /, 0 /, 0 /, -128 /, 2405 /, 3 /, 0 /
          DATA DB4 / 10000 /, 10000 /, 0 /, 0 /, -16 /, 2403 /, 3 /, 400 /
          DATA DB5 / 10000 /, 10000 /, 0 /, 0 /, 8 /, 2403 /, 3 /, 400 /
          DATA DB6 / 101000 /, 0 /, 0 /, 0 /, -16 /, 2405 /, 3 /, 400 /
          DATA DB7 / 140000 /, 20000 /, 0 /, 0 /, -256 /, 3003 /, 23 /, 400 /
          DATA DB8 / 40000 /, 20000 /, 0 /, 0 /, -256 /, 3005 /, 3 /, 400 /

```

```

FORTRAN IV      U01C-03      MON 29-OCT-79 14112:16      PAGE 004
CORE-08K, UIC-1.123J      ,TT0-DK3:CONLO1.FOR

0046      DATA DB9 / 110000 /, 60000 /, 0 /, 0 /, -256 /, 3003 /, 17 /, 0 /
0047      DATA DB10 / 10000 /, 160000 /, 0 /, 0 /, -256 /, 3005 /, 17 /, 0 /
0048      DATA DB11 / 0 /, 0 /, 0 /, 0 /, -1 /, 1403 /, 777 /, 1 /
0049      DATA DB12 / 0 /, 100000 /, 0 /, 0 /, -2 /, 3405 /, 777 /, 0 /

0050      DATA DB13 / 0 /, 0 /, 0 /, 0 /, -1 /, 3 /, 777 /, 1 /
0051      DATA DB14 / 0 /, 0 /, 0 /, 0 /, -1 /, 403 /, 3 /, 400 /
0052      DATA DB15 / 150000 /, 20000 /, 0 /, 0 /, -256 /, 3403 /, 0 /, 0 /
0053      DATA DB16 / 50000 /, 120000 /, 0 /, 0 /, -256 /, 3405 /, 0 /, 0 /
0054      DATA DB17 / 110000 /, 60000 /, 0 /, 0 /, -256 /, 3403 /, 0 /, 0 /
0055      DATA DB18 / 10000 /, 160000 /, 0 /, 0 /, -256 /, 3405 /, 0 /, 0 /
0056      DATA DB19 / 140000 /, 30000 /, 0 /, 0 /, -16 /, 3403 /, 0 /, 0 /
0057      DATA DB20 / 40000 /, 130000 /, 0 /, 0 /, -16 /, 3405 /, 0 /, 0 /

```

```

0052      DATA SPACE / /

          ;
          ; THE FOLLOWING TABLE DEFINES WHICH FUNCTIONS ARE
          ; FUNCTION 1,2,3,ETC.
          ;
          ; EG. FUNCTION 9 IS WFM - WRITE FUNCTION MEMORY

```

| FORTRAN IV         | STORAGE MAP                            |
|--------------------|----------------------------------------|
| NAME               | OFFSET    ATTRIBUTES                   |
| ICPROBE            | 000014 INTEGERx2 VARIABLE              |
| INNUMB             | 000070 INTEGERx2 VARIABLE              |
| SPACE              | 000016 LOGICALx1 VARIABLE              |
| <br>               |                                        |
| COMMON BLOCK /DDB/ | LENGTH 000500                          |
| DDB                | 000000 INTEGERx2 ARRAY (8,20) VECTORED |
| DB1                | 000000 INTEGERx2 ARRAY (8)             |
| DB2                | 000020 INTEGERx2 ARRAY (8)             |
| DB3                | 000040 INTEGERx2 ARRAY (8)             |
| DB4                | 000060 INTEGERx2 ARRAY (8)             |
| DB5                | 000100 INTEGERx2 ARRAY (8)             |
| DB6                | 000120 INTEGERx2 ARRAY (8)             |
| DB7                | 000140 INTEGERx2 ARRAY (8)             |
| DB8                | 000160 INTEGERx2 ARRAY (8)             |
| DB9                | 000200 INTEGERx2 ARRAY (8)             |
| DB10               | 000220 INTEGERx2 ARRAY (8)             |
| DB11               | 000240 INTEGERx2 ARRAY (8)             |
| DB12               | 000260 INTEGERx2 ARRAY (8)             |
| DB13               | 000300 INTEGERx2 ARRAY (8)             |
| DB14               | 000320 INTEGERx2 ARRAY (8)             |
| DB15               | 000340 INTEGERx2 ARRAY (8)             |
| DB16               | 000360 INTEGERx2 ARRAY (8)             |
| DB17               | 000400 INTEGERx2 ARRAY (8)             |
| DB18               | 000420 INTEGERx2 ARRAY (8)             |
| DB19               | 000440 INTEGERx2 ARRAY (8)             |
| DB20               | 000460 INTEGERx2 ARRAY (8)             |
| <br>               |                                        |
| COMMON BLOCK /TC/  | LENGTH 000044                          |

| COMMON BLOCK | COMVAR | LENGTH |
|--------------|--------|--------|
| TCUIN        | 000000 | 000000 |
| TCUIN        | 000002 | 000000 |
| TCUIN        | 000004 | 000000 |
| TCUIN        | 000006 | 000000 |
| TCUIN        | 000010 | 000000 |
| TCUIN        | 000012 | 000000 |
| TCUIN        | 000014 | 000000 |
| TCUIN        | 000016 | 000000 |
| TCUIN        | 000020 | 000000 |
| TCUIN        | 000022 | 000000 |
| TCUIN        | 000024 | 000000 |
| TCUIN        | 000026 | 000000 |
| TCUIN        | 000030 | 000000 |
| TCUIN        | 000032 | 000000 |
| TCUIN        | 000034 | 000000 |
| TCUIN        | 000036 | 000000 |
| TCUIN        | 000040 | 000000 |
| TCUIN        | 000042 | 000000 |

| COMMON BLOCK | DATA   | LENGTH |
|--------------|--------|--------|
| TCUIN        | 000000 | 000000 |
| TCUIN        | 000002 | 000000 |
| TCUIN        | 000004 | 000000 |
| TCUIN        | 000006 | 000000 |
| TCUIN        | 000010 | 000000 |
| TCUIN        | 000012 | 000000 |
| TCUIN        | 000014 | 000000 |
| TCUIN        | 000016 | 000000 |
| TCUIN        | 000020 | 000000 |
| TCUIN        | 000022 | 000000 |
| TCUIN        | 000024 | 000000 |
| TCUIN        | 000026 | 000000 |
| TCUIN        | 000030 | 000000 |
| TCUIN        | 000032 | 000000 |
| TCUIN        | 000034 | 000000 |
| TCUIN        | 000036 | 000000 |
| TCUIN        | 000040 | 000000 |
| TCUIN        | 000042 | 000000 |



**COMMON -DDB/ DDB**

COMMON /DATAA/ TCC

**END OF DATA DEFINITION**

NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CMOPEN

**THE UNIVERSITY OF CHICAGO PRESS**

**SET FUNCTION CODE**

**TCC-TCWIN**

CALL TRANSFER CODE 1 COMMON ROUTINE

IUM-ITC1 (IINO, IMLN, IBUF)

## RETURN

**RETURN  
END**

| NAME | OFFSET | ATTRIBUTES | STORAGE MAP |
|------|--------|------------|-------------|
|------|--------|------------|-------------|



```
COMMON BLOCK /DATA/  LENGTH 000002
TCC  000000  INTEGERx2 VARIABLE
```

FUNCTIONAL DESCRIPTION:

THIS FUNCTION IS USED TO READ ONE LINE OF DATA FROM THE COMTAL IMAGE SELECTED TO THE USER BUFFER.

PARAMETERS:

```

      INNO : THE IMAGE NUMBER (1-3)
      INLN : THE LINE NUMBER (1-256) 1 AT TOP, 256 AT BOTTOM
      BUFL : USER BUFFER ADDRESS

```

```

      ERRORS:
      :
      : 0 : SUCCESS
      : -1 : WRITE ERROR

```

```

      *****

```

# DATA DEFINITION

```

      INTEGER TCUIH, TCRIH, TCUGH, TCRGH, TCUPS, TCRPS
      INTEGER TCUFH, TCRFH, TCUTP, TCRTP, TCUDC, TCRDC
      INTEGER TCUCB, TCRCB, TCUCD, TCRCD, TCUCB, TCRDB

```

```

      INTEGER DDB(8,20)
      INTEGER TCC

```

```

      COMMON /DDB/ DDB

```

```

      COMMON /TC/ TCUIH, TCRIH, TCUGH, TCRGH, TCUPS, TCRPS,
      TCUFH, TCRFH, TCUTP, TCRTP, TCUDC, TCRDC,
      TCUCB, TCRCB, TCUCD, TCRCD, TCUCB, TCRDB

```

```

      COMMON /DATAA/ TCC
      END OF DATA DEFINITION

```

```

      NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CROPER
      -----

```

```

      FORTRAN IU      U01C-03      MON 29-OCT-79 14:14:05      PAGE 002
      CORE=08K, UIC=C1.123J      ,TT0=-DK3:CONL02.FOR

```

```

      *****

```

```

      , SET FUNCTION CODE

```

```

0010      TCC=TCRIM
0011      . CALL TRANSFER CODE 1 COMMON ROUTINE
0012      IRI=ITC1(IMNO,IMLN,IBUF)
0013      RETURN
      END

```

# FORTRAN IV STORAGE MAP

| NAME | OFFSET | ATTRIBUTES                   |
|------|--------|------------------------------|
| IRIM | 000022 | INTEGER12 VARIABLE           |
| IMNO | 000014 | INTEGER12 PARAMETER VARIABLE |
| IMLN | 000016 | INTEGER12 PARAMETER VARIABLE |
| IBUF | 000020 | INTEGER12 PARAMETER VARIABLE |
| ITC1 | 000000 | INTEGER12 PROCEDURE          |

COMMON BLOCK /DDB/ LENGTH 000500

DDE 000000 INTEGER12 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/ LENGTH 000044

|       |        |                    |
|-------|--------|--------------------|
| TCRIM | 000000 | INTEGER12 VARIABLE |
| TCRIM | 000002 | INTEGER12 VARIABLE |
| TCUCH | 000004 | INTEGER12 VARIABLE |
| TCUCH | 000006 | INTEGER12 VARIABLE |
| TCUPS | 000010 | INTEGER12 VARIABLE |
| TCUPS | 000012 | INTEGER12 VARIABLE |
| TCUFH | 000014 | INTEGER12 VARIABLE |
| TCUFH | 000016 | INTEGER12 VARIABLE |
| TCUTP | 000020 | INTEGER12 VARIABLE |
| TCUTP | 000022 | INTEGER12 VARIABLE |
| TCUDC | 000024 | INTEGER12 VARIABLE |
| TCUDC | 000026 | INTEGER12 VARIABLE |
| TCUCB | 000030 | INTEGER12 VARIABLE |
| TCUCB | 000032 | INTEGER12 VARIABLE |
| TCUCD | 000034 | INTEGER12 VARIABLE |
| TCUCD | 000036 | INTEGER12 VARIABLE |
| TCUDB | 000040 | INTEGER12 VARIABLE |
| TCUDB | 000042 | INTEGER12 VARIABLE |

COMMON BLOCK /DATA/ LENGTH 000002

TCC 000000 INTEGER12 VARIABLE



```

COMMON /DATAA/ TCC
END OF DATA DEFINITION
NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CMOPEN
-----

```

```

FORTRAN IV      U01C-03      MON 29-OCT-79 14:14:29      PAGE 002
CORE-08K, UTC-C1,123J      ,TTO:-DK3:CONL02.FOR

```

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
,   SET FUNCTION CODE
TCC-TCUGM
,   CALL TRANSFER CODE 1 COMMON ROUTINE
IUGM-ITC1(IMNO,IMLN,IBUF)
RETURN
END

```

```

FORTRAN IV      STORAGE MAP
NAME  OFFSET  ATTRIBUTES
IUGM  000022  INTEGER12 VARIABLE
IMHO  000014  INTEGER12 PARAMETER VARIABLE
IMLN  000016  INTEGER12 PARAMETER VARIABLE
IBUF  000020  INTEGER12 PARAMETER VARIABLE
ITC1  000000  INTEGER12 PROCEDURE
COMMON BLOCK /DDB/      LENGTH 000500
DDB   000000  INTEGER12 ARRAY (8,20) VECTORED
COMMON BLOCK /TC/      LENGTH 000044

```

|        |           |          |
|--------|-----------|----------|
| 000000 | INTECDP12 | 0000AB1E |
| 000002 | INTECDP12 | 0000AB1E |
| 000004 | INTECDP12 | 0000AB1E |
| 000006 | INTECDP12 | 0000AB1E |
| 000008 | INTECDP12 | 0000AB1E |
| 000010 | INTECDP12 | 0000AB1E |
| 000012 | INTECDP12 | 0000AB1E |
| 000014 | INTECDP12 | 0000AB1E |
| 000016 | INTECDP12 | 0000AB1E |
| 000018 | INTECDP12 | 0000AB1E |
| 000020 | INTECDP12 | 0000AB1E |
| 000022 | INTECDP12 | 0000AB1E |
| 000024 | INTECDP12 | 0000AB1E |
| 000026 | INTECDP12 | 0000AB1E |
| 000030 | INTECDP12 | 0000AB1E |
| 000032 | INTECDP12 | 0000AB1E |
| 000034 | INTECDP12 | 0000AB1E |
| 000036 | INTECDP12 | 0000AB1E |
| 000038 | INTECDP12 | 0000AB1E |
| 000040 | INTECDP12 | 0000AB1E |
| 000042 | INTECDP12 | 0000AB1E |

200000 M.L.M.T. / MAY 1970 - 1365.8 M.L.M.T.

08-06-78 14:15:00  
MON 29-DEC-78 14:15:00  
PAGE 00:  
CMT:CM:02.F00

6622: INTERFER. FUNCTION: BGRN (PWR): 15.14, 18.45

FACT: CAGAL DESCRIP: ISM:

THIS FUNCTION IS USED TO READ ONE LINE OF DATA FROM THE CONTROL GRAPHIC SELECTED TO THE USER BUFFER.

15931300000

```

      .
      .
      .
      . THE LOGICAL NUMBER 1-3,
      . USER TIME NUMBER 1-256, ; AT TOP, 256 AT BOTTOM
      . USER BUFFER ADDRESS

```

50943

```

      0 : SUCCESS
      -1 : WRITE ERROR

```

FORTRAN IV U01C-03 PAGE 002  
CONF-08K. UIC-51,123J .TT8:-DK3:COML02.FOR

```
00000000C      C      C  
0010          .    SET FUNCTION CODE  
TCC-TCRCH  
0011          .    CALL TRANSFER CODE 1 COMMON ROUTINE  
IRCA-ITC1(INNO,IMLN,IBUF)  
RETURN  
END  
0012  
0013
```

| FORTRAN IU                         |        | STORAGE MAP            |          |
|------------------------------------|--------|------------------------|----------|
| NAME                               | OFFSET | ATTRIBUTES             |          |
| IRCM                               | 000022 | INTEGER12 VARIABLE     |          |
| IRNO                               | 000014 | INTEGER12 PARAMETER    | VARIABLE |
| IRLN                               | 000016 | INTEGER12 PARAMETER    | VARIABLE |
| IRUF                               | 000020 | INTEGER12 PARAMETER    | VARIABLE |
| ITC1                               | 000000 | INTEGER12 PROCEDURE    |          |
| COMMON BLOCK /DDB/ LENGTH 000500   |        |                        |          |
| DDB                                | 000000 | INTEGER12 ARRAY (8,20) | VECTORED |
| COMMON BLOCK /TC/ LENGTH 000044    |        |                        |          |
| TCUIM                              | 000000 | INTEGER12 VARIABLE     |          |
| TCUIM                              | 000002 | INTEGER12 VARIABLE     |          |
| TCUGM                              | 000004 | INTEGER12 VARIABLE     |          |
| TCUGM                              | 000006 | INTEGER12 VARIABLE     |          |
| TCUPS                              | 000010 | INTEGER12 VARIABLE     |          |
| TCRPS                              | 000012 | INTEGER12 VARIABLE     |          |
| TCUFM                              | 000014 | INTEGER12 VARIABLE     |          |
| TCUFM                              | 000016 | INTEGER12 VARIABLE     |          |
| TCUTP                              | 000020 | INTEGER12 VARIABLE     |          |
| TCUTP                              | 000022 | INTEGER12 VARIABLE     |          |
| TCWDC                              | 000024 | INTEGER12 VARIABLE     |          |
| TCWDC                              | 000026 | INTEGER12 VARIABLE     |          |
| TCUCB                              | 000030 | INTEGER12 VARIABLE     |          |
| TCUCB                              | 000032 | INTEGER12 VARIABLE     |          |
| TCUCD                              | 000034 | INTEGER12 VARIABLE     |          |
| TCUCD                              | 000036 | INTEGER12 VARIABLE     |          |
| TCUCB                              | 000040 | INTEGER12 VARIABLE     |          |
| TCUCB                              | 000042 | INTEGER12 VARIABLE     |          |
| COMMON BLOCK /DATAA/ LENGTH 000002 |        |                        |          |
| TCC                                | 000000 | INTEGER12 VARIABLE     |          |





FORTRAN II U01C-03 MON 29-OCT-79 14:15:22 PAGE 002  
 COME-OSK. UIC-E1.1233 ,TT01-DK31CONL02.FOR

```

0000 COMMON /DATAA/ TCC
0000 END OF DATA DEFINITION
0000 NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CROPEM
0000 -----
0000 *****
0000 CORRECT IMAGE NUMBER, CORRECT LINE NUMBER
0010 NO-IMNO-1
0011 LN-IMLN-1
0012 MAXLN-256
0013 MAXGM-2
0014 MAXIM-2
0015 FORCE IMAGE/GRAPHIC NUMBER, 0 <= IMAGE/GRAPHIC NO. < 256
0016 IF((NO.GT.MAXIM).OR.(NO.LT.0)) NO=0
    IF((LN.GT.MAXLN).OR.(LN.LT.0)) LN=0
    SET ITC1=SUCCESS
0019 ITC1=0
    GET USER BUFFER ADDRESS IN DDB
    DDB(4,TCC)=IADDR(1BUF)
    SET TRANSFER COMMAND: SET IMAGE NUMBER, ADD LINE NUMBER
    DDB(3,TCC)=NO*4000
    DDB(3,TCC)=DDB(3,TCC)+LN
    CALL HANDLER
    IC=IUPITU(DDB(1,TCC))
    CHECK FOR ERROR
    IF(IC.LT.0) ITC1=IC
    RETURN
0024 END
0026
0027

```

| FORTRAN I/O |        | STORAGE MAP                  |
|-------------|--------|------------------------------|
| NAME        | OFFSET | ATTRIBUTES                   |
| ITC1        | 000022 | INTEGER12 VARIABLE           |
| ITN0        | 000014 | INTEGER12 PARAMETER VARIABLE |
| ITLN        | 000016 | INTEGER12 PARAMETER VARIABLE |
| IEUF        | 000020 | INTEGER12 PARAMETER VARIABLE |
| NO          | 000074 | INTEGER12 VARIABLE           |
| LN          | 000076 | INTEGER12 VARIABLE           |
| MAXLN       | 000100 | INTEGER12 VARIABLE           |
| MAXGN       | 000102 | INTEGER12 VARIABLE           |
| MAXIN       | 000104 | INTEGER12 VARIABLE           |
| IADDR       | 000000 | INTEGER12 PROCEDURE          |
| IC          | 000106 | INTEGER12 VARIABLE           |
| IURITU      | 000000 | INTEGER12 PROCEDURE          |

| COMMON BLOCK /DOB/ |        | LENGTH 000500                   |
|--------------------|--------|---------------------------------|
| DOB                | 000000 | INTEGER12 ARRAY (8,20) VECTORED |

| COMMON BLOCK /TC/ |        | LENGTH 000044      |
|-------------------|--------|--------------------|
| TCUIM             | 000000 | INTEGER12 VARIABLE |
| TCUIM             | 000002 | INTEGER12 VARIABLE |
| TCUGM             | 000004 | INTEGER12 VARIABLE |
| TCUGM             | 000006 | INTEGER12 VARIABLE |
| TCUPS             | 000010 | INTEGER12 VARIABLE |
| TCUPS             | 000012 | INTEGER12 VARIABLE |
| TCUFM             | 000014 | INTEGER12 VARIABLE |
| TCUFM             | 000016 | INTEGER12 VARIABLE |
| TCUTP             | 000020 | INTEGER12 VARIABLE |
| TCUTP             | 000022 | INTEGER12 VARIABLE |
| TCUDC             | 000024 | INTEGER12 VARIABLE |
| TCUDC             | 000026 | INTEGER12 VARIABLE |
| TCUCB             | 000030 | INTEGER12 VARIABLE |
| TCUCB             | 000032 | INTEGER12 VARIABLE |
| TCUCD             | 000034 | INTEGER12 VARIABLE |
| TCUCD             | 000036 | INTEGER12 VARIABLE |
| TCUDE             | 000040 | INTEGER12 VARIABLE |
| TCUDE             | 000042 | INTEGER12 VARIABLE |

| COMMON BLOCK /DATAA/ |        | LENGTH 000002      |
|----------------------|--------|--------------------|
| TCC                  | 000000 | INTEGER12 VARIABLE |

FOR/

FORTRAN IV U01C-03 MON 29-OCT-79 14:16:39 PAGE 001  
 COME-QSA, UIC-C1,1233 ,TT0:-DK3:CONL03.FOR

FILE NAME: CONL03.FOR

DATE LAST EDITED: 21-MAR-79

HISTORY:

DATE: MODIFICATION

28-FEB-79 INITIAL WRITING  
 19-MAR-79 CHANGE FUNCTIONS  
 21-MAR-79 TO START WITH I  
 TO 'IFMNO',  
 TO 'IFMNO'

0001 INTEGER FUNCTION IUFN(IFMNO,IBUF)

FUNCTIONAL DESCRIPTION:

THIS ROUTINE WRITES THE SELECTED FUNCTION  
 MEMORY FROM THE USER IBUF.

PARAMETERS:

IFMNO = FUNCTION MEMORY NUMBER (1-3)  
 IBUF = USER IBUF (INTEGER ARRAY 256)

ERRORS:

0 = SUCCESS  
 -1 = WRITE ERROR

\*\*\*\*\*

DATA DEFINITION

INTEGER TCUMR, TCURM, TCURH, TCURM, TCURM, TCURM, TCURM, TCURM  
 INTEGER TCUMF, TCURF, TCURP, TCURP, TCURP, TCURP, TCURP, TCURP  
 INTEGER TCUCB, TCURB, TCURD, TCURD, TCURD, TCURD, TCURD, TCURD

0005 INTEGER TCC



COMMON BLOCK /DATAA/ LENGTH 000002 .

FORTRAN IV U01C-03 PAGE 001  
CORE-98K, UIC-11.1233 MON 29-OCT-79 14:17:02  
TT0:=DK3:COMLOS.FOR

```
0001      INTEGER FUNCTION IFRN(IFMNO,IBUF)
0002      FUNCTIONAL DESCRIPTION:
0003          THIS ROUTINE READS THE SELECTED FUNCTION
0004          MEMORY AND FILLS THE USER IBUFR.
0005
0006      PARAMETERS:
0007          IFMNO - FUNCTION MEMORY NUMBER (1-3)
0008          IBUF  - USER IBUFR (INTEGER ARRAY 256)
0009
0010      ERRORS:
0011          0 - SUCCESS
0012          -1 - WRITE ERROR
```

DATA DEFINITION

|      |                |        |        |        |        |       |
|------|----------------|--------|--------|--------|--------|-------|
| 0002 | INTEGER TCUIN, | TCRIN, | TUGH,  | TCRGN, | TCUPS, | TCRPS |
| 0003 | INTEGER TCUFN, | TCRFN, | TCUTP, | TCUDC, | TCUDC, | TCRDC |
| 0004 | INTEGER TCUCB, | TCRCB, | TCUCD, | TCUCB, | TCUCB, | TCRDB |

FORTRAN IU U01C-03 MON 29-OCT-79 14:17:02 PAGE 002  
CORE-08K, UIC-11,1233 ,TTO:-DK3:CONLO3.FOR

```

C      C
C      C      1) SET FUNCTION NUMBER
C      C      2) CALL TRANSFER CODE 2 COMMON ROUTINE
C      C
C      C      TCC-TCRPM
C      C      IRFM-ITC2(IFMNO,IBUF)
C      C
C      C      RETURN
C      C      END
0011
0012
0013
0014
```

| FORTRAN IV | STORAGE MAP |            |
|------------|-------------|------------|
| NAME       | OFFSET      | ATTRIBUTES |

```

      000020 INTEGER12 VARIABLE
      000014 INTEGER12 PARAMETER VARIABLE
      000016 INTEGER12 PARAMETER VARIABLE
      000020 INTEGER12 PROCEDURE

COMMON BLOCK /DDB/      LENGTH 000500

DDB  000000 INTEGER12 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/      LENGTH 000044

TCUIM 000000 INTEGER12 VARIABLE
TCUIM 000002 INTEGER12 VARIABLE
TCUGA 000004 INTEGER12 VARIABLE
TCUGA 000006 INTEGER12 VARIABLE
TCUPS 000010 INTEGER12 VARIABLE
TCRPS 000012 INTEGER12 VARIABLE
TCUFH 000014 INTEGER12 VARIABLE
TCUFH 000016 INTEGER12 VARIABLE
TCUTP 000020 INTEGER12 VARIABLE
TCUTP 000022 INTEGER12 VARIABLE
TCUDC 000024 INTEGER12 VARIABLE
TCUDC 000026 INTEGER12 VARIABLE
TCUCB 000030 INTEGER12 VARIABLE
TCUCB 000032 INTEGER12 VARIABLE
TCUCD 000034 INTEGER12 VARIABLE
TCUCD 000036 INTEGER12 VARIABLE
TCUDB 000040 INTEGER12 VARIABLE
TCUDB 000042 INTEGER12 VARIABLE

COMMON BLOCK /CONVAR/  LENGTH 000006

MAXIM 000000 INTEGER12 VARIABLE
MAXIM 000002 INTEGER12 VARIABLE
MAXIM 000004 INTEGER12 VARIABLE

COMMON BLOCK /DATAA/  LENGTH 000002

TCC  000000 INTEGER12 VARIABLE

```

FORTRAN IV    U01C-03    MON 29-OCT-79 14:17:24    PAGE 001  
 CORE-08K, UIC-E1,1233    ,TT0:=DK3:COML03.FOR

```

0001  C      INTEGER FUNCTION IUPS(IBUF)
      C      FUNCTIONAL DESCRIPTION:
      C      THIS ROUTINE WRITES THE PSEUDO-COLOR
      C      MEMORY FROM THE USER IBUF.

```





```

C      1) SET FUNCTION NUMBER
C      2) CALL TRANSFER CODE 2 COMMON ROUTINE
C
Q011  C      TCC=TCUPS
Q012  C      IUPS=ITC2(IFMNO,IBUF)
Q013  C      RETURN
Q014  C      END

```

# FORTRAN IU STORAGE MAP

## NAME OFFSET ATTRIBUTES

```

IUPS  000016  INTEGER*2 VARIABLE
IBUF  000014  INTEGER*2 PARAMETER VARIABLE
ITC2  000000  INTEGER*2 PROCEDURE
IFMNO 000070  INTEGER*2 VARIABLE

```

COMMON BLOCK /DDB/ LENGTH 000500

IDB 000000 INTEGER\*2 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/ LENGTH 000044

```

TCUIM 000000  INTEGER*2 VARIABLE
TCUIM 000002  INTEGER*2 VARIABLE
TCUGM 000004  INTEGER*2 VARIABLE
TCUGM 000006  INTEGER*2 VARIABLE
TCUPS 000010  INTEGER*2 VARIABLE
TCUPS 000012  INTEGER*2 VARIABLE
TCUFM 000014  INTEGER*2 VARIABLE
TCUFM 000016  INTEGER*2 VARIABLE
TCUTP 000020  INTEGER*2 VARIABLE
TCUTP 000022  INTEGER*2 VARIABLE
TCUDC 000024  INTEGER*2 VARIABLE
TCUDC 000026  INTEGER*2 VARIABLE
TCUCB 000030  INTEGER*2 VARIABLE
TCUCB 000032  INTEGER*2 VARIABLE
TCUCL 000034  INTEGER*2 VARIABLE
TCUCD 000036  INTEGER*2 VARIABLE
TCUDB 000040  INTEGER*2 VARIABLE
TCUDB 000042  INTEGER*2 VARIABLE

```

COMMON BLOCK /COMPAR/ LENGTH 000006

```

MAXIM 000000  INTEGER*2 VARIABLE
MAXLN 000002  INTEGER*2 VARIABLE
MAXGM 000004  INTEGER*2 VARIABLE

```

COMMON BLOCK /DATA/ LENGTH 000002  
 TCC INTEGER\*2 VARIABLE

FORTRAN IV U01C-03 MON 29-OCT-79 14:17:56 PAGE 001  
 CORE-08A, UIC-01,1233 ,TT0:-DK3:CONL03.FOR

```

0001 C      INTEGER FUNCTION IRPS(IBUF)
      C      FUNCTIONAL DESCRIPTION:
      C      THIS ROUTINE READS THE PSEUDO-COLOR
      C      MEMORY AND PUTS IT IN THE USER IBUF.
      C      PARAMETERS:
      C      IBUF - USER IBUF (INTEGER ARRAY 256)
      C      ERRORS:
      C      0 - SUCCESS
      C      -1 - WRITE ERROR
      C      *****
      C      DATA DEFINITION
      C      *****
      C      INTEGER TCUIH, TCRIM, TCUGH, TCRGM, TCUPS, TCRPS
      C      INTEGER TCUFH, TCRFH, TCUTP, TCRTP, TCUDC, TCRDC
      C      INTEGER TCUCB, TCRCB, TCUCD, TCRCD, TCUDB, TCRDB
      C      INTEGER DBS(8,20)
      C      INTEGER TCC
      C      COMMON /DBS/ DBS
      C      COMMON /TC/ TCUIH, TCRIM, TCUGH, TCRGM, TCUPS, TCRPS,
      C      TCUFH, TCRFH, TCUTP, TCRTP, TCUDC, TCRDC,
      C      TCUCB, TCRCB, TCUCD, TCRCD, TCUDB, TCRDB
      C      COMMON /DATA/ TCC
  
```



AD-A084 895

MEASUREMENT CONCEPT CORP ROME NY F/8 17/9  
TACTICAL TARGET IDENTIFICATION (TTI) LABORATORY SIMULATION. VOL--ETC(U)  
MAR 80 R KELLOGG, R A MCKIE, J TIMS F30602-79-C-0029

RADC -TR-80-41-VOL-1

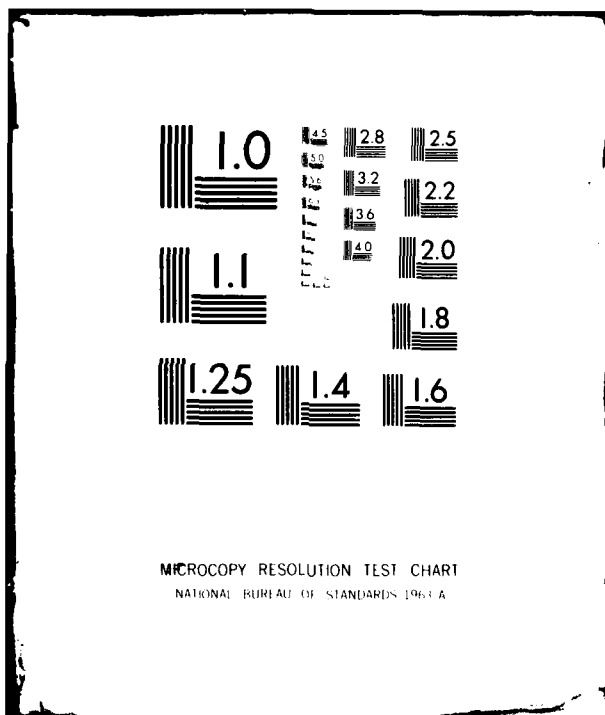
NL

UNCLASSIFIED

3-43

AL  
1/8-49/8

END  
DATE  
FILMED  
7 80  
DTIC



```

000012 INTEGER*2 VARIABLE
000014 INTEGER*2 VARIABLE
000016 INTEGER*2 VARIABLE
000020 INTEGER*2 VARIABLE
000022 INTEGER*2 VARIABLE
000024 INTEGER*2 VARIABLE
000026 INTEGER*2 VARIABLE
000030 INTEGER*2 VARIABLE
000032 INTEGER*2 VARIABLE
000034 INTEGER*2 VARIABLE
000036 INTEGER*2 VARIABLE
000040 INTEGER*2 VARIABLE
000042 INTEGER*2 VARIABLE

COMMON BLOCK /DATA/ LENGTH 000002

TCC 000000 INTEGER*2 VARIABLE

```

FORTRAN IU U01C-03 MON 29-OCT-79 14:18:18 PAGE 001  
CORE-08K, UIC-E1,1233 ,TT0:-DK3:COM103.FOR

```

0001 C INTEGER FUNCTION ITCE(IFMNO,IBUF)
C
C FUNCTIONAL DESCRIPTION:
C THIS ROUTINE IS CALLED BY FUNCTION MEMORY , AND PSEUDO
C COLOR MEMORY, READ/WRITE
C
C CALLER SETS THE FUNCTION NUMBER
C THE ROUTINE:
C 1) DETERMINES WHETHER PSEUDO OR FUNCTION (ONLY 1 PSEUDO MEMORY)
C 2) FORCES FUNCTION MEMORY < 4
C 3) SETS COMMAND FOR TRANSFER CODE 2
C 4) CALLS HANDLER
C
C PARAMETERS:
C IFMNO - FUNCTION MEMORY NUMBER (1-3)
C IBUF - USER IBUFFER (INTEGER ARRAY 256)
C
C ERRORS:
C 0 - SUCCESS
C -1 - WRITE ERROR

```

\*\*\*\*\*

DATA DEFINITION

INTEGER TCUM, TCRIN, TCRH, TCRN, TCPS, TCPS  
INTEGER TCUPH, TCUPM, TCUTP, TCUDC, TCUDC,  
INTEGER TCUCB, TCRCB, TCUCD, TCRCB, TCUCD, TCRCB

INTEGER DBS(8,20)  
INTEGER TCC

COMMON /DBS/ DBS

COMMON /TC/ TCUM, TCRIN, TCRH, TCRN, TCPS, TCPS,  
TCUPH, TCUPM, TCUTP, TCUDC, TCUDC, TCUDC,  
TCUCB, TCRCB, TCUCD, TCRCB, TCUCD, TCRCB

COMMON /CONVAR/ MAXIN, MAXLN, MAXGM

FORTRAN IV UIC-93 MON 29-OCT-79 14:18:18 PAGE 002  
CORE=98K, UIC-E1,1233 ,TTE:-DK3:COML03.FOR

0010 COMMON /DATAA/ TCC

END OF DATA DEFINITION

NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CROPEN

\*\*\*\*\*

SET SUCCESS

ITC2=0

CHECK FOR PSEUDO COLOR MEMORY

IF((TCC.EQ.TCPS).OR.(TCC.EQ.TCPS)) GO TO 100

FUNCTION MEMORY

CORRECT FUNCTION MEMORY NUMBER



```

C
0014 LIMIT FUNCTION MEMORY NUMBER < 4
0015 NO-IFNO-1
      IF (NO.GT.MAXGN).OR.(NO.LT.0)) NO=0
C
0017 SET FUNCTION MEMORY NUMBER IN TRANSFER COMMAND
      DDB(3,TCC)=NO*400
C
0018 GO TO 200
C
0019 PSEUDO COLOR MEMORY
      DDB(3,TCC)=0
C
0020 SET IBUF ADDRESS
      DDB(4,TCC)=IADDR(IBUF)
C
0021 CALL HANDLER
      IC=IURITU(DDB(1,TCC))
C
0022 CHECK FOR ERROR
      IF (IC.LT.0) ITC2=IC
C
0024 RETURN
0025 END

```

| FORTRAN I/O                           |        | STORAGE MAP                     |
|---------------------------------------|--------|---------------------------------|
| NAME                                  | OFFSET | ATTRIBUTES                      |
| ITC2                                  | 000020 | INTEGER*2 VARIABLE              |
| IFNO                                  | 000014 | INTEGER*2 PARAMETER VARIABLE    |
| IBUF                                  | 000016 | INTEGER*2 PARAMETER VARIABLE    |
| IN                                    | 000072 | INTEGER*2 VARIABLE              |
| IADDR                                 | 000000 | INTEGER*2 PROCEDURE             |
| IC                                    | 000074 | INTEGER*2 VARIABLE              |
| IURITU                                | 000000 | INTEGER*2 PROCEDURE             |
| COMMON BLOCK /DDB/      LENGTH 000500 |        |                                 |
| DDB                                   | 000000 | INTEGER*2 ARRAY (8,20) VECTORED |
| COMMON BLOCK /TC/      LENGTH 000044  |        |                                 |
| TCUIM                                 | 000000 | INTEGER*2 VARIABLE              |
| TCRIM                                 | 000002 | INTEGER*2 VARIABLE              |

|                                     |        |           |          |
|-------------------------------------|--------|-----------|----------|
| TCUW                                | 000004 | INTEGER32 | VARIABLE |
| TCUW                                | 000006 | INTEGER32 | VARIABLE |
| TCUW                                | 000010 | INTEGER32 | VARIABLE |
| TCUW                                | 000012 | INTEGER32 | VARIABLE |
| TCUW                                | 000014 | INTEGER32 | VARIABLE |
| TCUW                                | 000016 | INTEGER32 | VARIABLE |
| TCUW                                | 000020 | INTEGER32 | VARIABLE |
| TCUW                                | 000022 | INTEGER32 | VARIABLE |
| TCUW                                | 000024 | INTEGER32 | VARIABLE |
| TCUW                                | 000026 | INTEGER32 | VARIABLE |
| TCUW                                | 000030 | INTEGER32 | VARIABLE |
| TCUW                                | 000032 | INTEGER32 | VARIABLE |
| TCUW                                | 000034 | INTEGER32 | VARIABLE |
| TCUW                                | 000036 | INTEGER32 | VARIABLE |
| TCUW                                | 000040 | INTEGER32 | VARIABLE |
| TCUW                                | 000042 | INTEGER32 | VARIABLE |
| COMMON BLOCK /CONVAR/ LENGTH 000006 |        |           |          |
| MAXIM                               | 000000 | INTEGER32 | VARIABLE |
| MAXLN                               | 000002 | INTEGER32 | VARIABLE |
| MAXGM                               | 000004 | INTEGER32 | VARIABLE |
| COMMON BLOCK /DATA/ LENGTH 000002   |        |           |          |
| TCC                                 | 000000 | INTEGER32 | VARIABLE |
| FOR>                                |        |           |          |

```

FORTRAN IV          UIC-03          MON 29-OCT-79 14:19:51          PAGE 001
CORE-08K, UIC-C1.123J          ,IT0-DK3:CONL04.FOR
                                FILE NAME: CONL04.FOR
                                DATE LAST EDITED: 28-FEB-79
                                HISTORY:
                                DATE:          MODIFICATION
                                28-FEB-79      INITIAL WRITING

                                INTEGER FUNCTION UTP(IX,IY,IXV)
                                FUNCTIONAL DESCRIPTION:
                                THIS ROUTINE WRITES X, OR Y, OR BOTH, TO THE
                                TARGET CURSOR. (BOTH ACTUALLY REQUIRES 2 TRANSFERS)

                                PARAMETERS:
                                IX  - X COORDINATE OF CURSOR (1-256)
                                IY  - Y COORDINATE OF CURSOR (1-256)
                                IXV - 0 TRANSFER BOTH X AND Y
                                       1 TRANSFER X ONLY
                                       2 TRANSFER Y ONLY

                                ERRORS:
                                0  - SUCCESS
                                -1 - WRITE ERROR

                                *****
                                DATA DEFINITION
                                INTEGER TCUIH, TCRIH, TCUMH, TCRGH, TCUPS, TCAPS
                                INTEGER TCUFH, TCRFH, TCUTP, TCNDC, TCRDC
                                INTEGER TCUCB, TCRCB, TCUCD, TCRDB, TCRDB

                                INTEGER DBB(8,20)
                                INTEGER TCC

```



```

      CALL HANDLER
      IC=IURITU(DD8(1,TCC))
      CHECK FOR ERROR
      IF(IC.LT.0) UTP=IC
      RETURN
      DD8(3,TCC)=IX

```

FORTRAN IV    V01C-03    MON 29-OCT-79 14:19:51    PAGE 003  
 CORE-08K, UIC-E1,123J    .TTO=DK3:CONL04.FOR

```

      IC=IURITU(DD8(1,TCC))
      DD8(3,TCC)=IX
      ID=IURITU(DD8(1,TCC))
      CHECK ERROR
      IF(IC.LT.0) UTP=IC
      IF(ID.LT.0) UTP=ID
      RETURN
      END

```

| FORTRAN IV |        | STORAGE MAP |                    |
|------------|--------|-------------|--------------------|
| NAME       | OFFSET | ATTRIBUTES  |                    |
| UTP        | 000022 | INTEGER12   | VARIABLE           |
| IX         | 000014 | INTEGER12   | PARAMETER VARIABLE |
| IY         | 000016 | INTEGER12   | PARAMETER VARIABLE |
| IC         | 000020 | INTEGER12   | PARAMETER VARIABLE |
| ID         | 000074 | INTEGER12   | VARIABLE           |
| IURITU     | 000000 | INTEGER12   | PROCEDURE          |
| ID         | 000076 | INTEGER12   | VARIABLE           |

COMMON BLOCK /DD8/    LENGTH 000F00

D3F 000000 INTEGER12 ARRAY (8,20) VECTORED  
COMMON BLOCK /TC/ LENGTH 000044

TCUIM 000000 INTEGER12 VARIABLE  
TCUIM 000002 INTEGER12 VARIABLE  
TCUIM 000004 INTEGER12 VARIABLE  
TCUIM 000006 INTEGER12 VARIABLE  
TCUIM 000008 INTEGER12 VARIABLE  
TCUIM 000010 INTEGER12 VARIABLE  
TCUIM 000012 INTEGER12 VARIABLE  
TCUIM 000014 INTEGER12 VARIABLE  
TCUIM 000016 INTEGER12 VARIABLE  
TCUIM 000018 INTEGER12 VARIABLE  
TCUIM 000020 INTEGER12 VARIABLE  
TCUIM 000022 INTEGER12 VARIABLE  
TCUIM 000024 INTEGER12 VARIABLE  
TCUIM 000026 INTEGER12 VARIABLE  
TCUIM 000028 INTEGER12 VARIABLE  
TCUIM 000030 INTEGER12 VARIABLE  
TCUIM 000032 INTEGER12 VARIABLE  
TCUIM 000034 INTEGER12 VARIABLE  
TCUIM 000036 INTEGER12 VARIABLE  
TCUIM 000038 INTEGER12 VARIABLE  
TCUIM 000040 INTEGER12 VARIABLE  
TCUIM 000042 INTEGER12 VARIABLE

COMMON BLOCK /DATA/ LENGTH 000062

TCC 000000 INTEGER12 VARIABLE

FORTPAN IU V01C-03 MON 29-OCT-79 14:20:25 PAGE 001  
CORE-08K, UTC-C1,123J ,TTO:=DK3:CONLO4.FOR

0001 C INTEGER FUNCTION RTP(IX,IV)

C FUNCTIONAL DESCRIPTION:

C THIS ROUTINE READS THE X AND Y POSITION OF THE TARGET  
C AND PUTS THE COORDINATES IN IX, AND IV .

C PARAMETERS:

C IX : X POSITION OF TARGET (1-256)  
C IV : Y POSITION OF TARGET (1-256)

C ERRORS:

C 0 : SUCCESS  
C -1 : WRITE ERROR

C \*\*\*\*\*

```

0002      DATA DEFINITION
0003      INTEGER TMR(2)
0004
0005      INTEGER TCNIN, TCRIM, TCUNH, TCRGM, TCUPS, TCRPS,
0006      INTEGER TCNFM, TCRFM, TCUTP, TCUDC, TCRDC,
0007      INTEGER TCUCB, TCRCB, TCRCD, TCUDB, TCRDB
0008
0009      INTEGER DDB(8,20)
0010      INTEGER TCC
0011
0012      COMMON /DDB/ DDB
0013
0014      COMMON /TC/ TCNIN, TCRIM, TCUNH, TCRGM, TCUPS, TCRPS,
0015      TCNFM, TCRFM, TCUTP, TCUDC, TCRDC,
0016      TCUCB, TCRCB, TCRCD, TCUDB, TCRDB
0017
0018      COMMON /DATA/ TCC
0019
0020      END OF DATA DEFINITION
0021
0022      NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CROPER
0023      -----
0024      *****
0025      SET SUCCESS
0026      RTP=0
0027
0028
0029
0030
0031
0032
0033
0034
0035
0036
0037
0038
0039
0040
0041
0042
0043
0044
0045
0046
0047
0048
0049
0050
0051
0052
0053
0054
0055
0056
0057
0058
0059
0060
0061
0062
0063
0064
0065
0066
0067
0068
0069
0070
0071
0072
0073
0074
0075
0076
0077
0078
0079
0080
0081
0082
0083
0084
0085
0086
0087
0088
0089
0090
0091
0092
0093
0094
0095
0096
0097
0098
0099
0100
0101
0102
0103
0104
0105
0106
0107
0108
0109
0110
0111
0112
0113
0114
0115
0116
0117
0118
0119
0120
0121
0122
0123
0124
0125
0126
0127
0128
0129
0130
0131
0132
0133
0134
0135
0136
0137
0138
0139
0140
0141
0142
0143
0144
0145
0146
0147
0148
0149
0150
0151
0152
0153
0154
0155
0156
0157
0158
0159
0160
0161
0162
0163
0164
0165
0166
0167
0168
0169
0170
0171
0172
0173
0174
0175
0176
0177
0178
0179
0180
0181
0182
0183
0184
0185
0186
0187
0188
0189
0190
0191
0192
0193
0194
0195
0196
0197
0198
0199
0200

```

FORTRAN IV U01C-03 MON 29-OCT-79 14:20:25 PAGE 002  
 CORE=08K, UIC=E1,123] ,TTE=DK3:COML04.FOR

```

0012      SET FUNCTION = READ TARGET POSITION
0013      TCC=TCRTP
0014      GET ADDRESS
0015      DDB(4,TCC)=IADDR(TAR)
0016      CALL HANDLER
0017      IC=IURITU(DDB(1,TCC))
0018
0019
0020
0021
0022
0023
0024
0025
0026
0027
0028
0029
0030
0031
0032
0033
0034
0035
0036
0037
0038
0039
0040
0041
0042
0043
0044
0045
0046
0047
0048
0049
0050
0051
0052
0053
0054
0055
0056
0057
0058
0059
0060
0061
0062
0063
0064
0065
0066
0067
0068
0069
0070
0071
0072
0073
0074
0075
0076
0077
0078
0079
0080
0081
0082
0083
0084
0085
0086
0087
0088
0089
0090
0091
0092
0093
0094
0095
0096
0097
0098
0099
0100
0101
0102
0103
0104
0105
0106
0107
0108
0109
0110
0111
0112
0113
0114
0115
0116
0117
0118
0119
0120
0121
0122
0123
0124
0125
0126
0127
0128
0129
0130
0131
0132
0133
0134
0135
0136
0137
0138
0139
0140
0141
0142
0143
0144
0145
0146
0147
0148
0149
0150
0151
0152
0153
0154
0155
0156
0157
0158
0159
0160
0161
0162
0163
0164
0165
0166
0167
0168
0169
0170
0171
0172
0173
0174
0175
0176
0177
0178
0179
0180
0181
0182
0183
0184
0185
0186
0187
0188
0189
0190
0191
0192
0193
0194
0195
0196
0197
0198
0199
0200

```

```

C
      0015      CHECK FOR ERRORS
      0016      IF (IC.LT.0) RTP=IC
      0017      SET X, AND Y (4097 IS OFFSET FROM 1)
      0018      IX=TAR(1)+4097
      0019      IY=TAR(2)+4097
      0020      RETURN
      0021      END

```

```

FORTRAN IV      STORAGE MAP
NAME      OFFSET      ATTRIBUTES
TAR      000020      INTEGER12 ARRAY (2)
RTP      000024      INTEGER12 VARIABLE
IX      000014      INTEGER12 PARAMETER VARIABLE
IY      000016      INTEGER12 PARAMETER VARIABLE
IADDR      000000      INTEGER12 PROCEDURE
IC      000076      INTEGER12 VARIABLE
IURITU      000000      INTEGER12 PROCEDURE

```

```

COMMON BLOCK /DDB/      LENGTH 000500
DDE      000000      INTEGER12 ARRAY (8,20) VECTORED

```

```

COMMON BLOCK /TC/      LENGTH 000044

```

```

TCUIM      000000      INTEGER12 VARIABLE
TCPIM      000002      INTEGER12 VARIABLE
TCUGM      000004      INTEGER12 VARIABLE
TCRCM      000006      INTEGER12 VARIABLE
TCUPS      000010      INTEGER12 VARIABLE
TCRPS      000012      INTEGER12 VARIABLE
TCUFM      000014      INTEGER12 VARIABLE
TCPFM      000016      INTEGER12 VARIABLE
TCMTP      000020      INTEGER12 VARIABLE
TCMTP      000022      INTEGER12 VARIABLE
TCUDC      000024      INTEGER12 VARIABLE
TCRDC      000026      INTEGER12 VARIABLE
TCUCB      000030      INTEGER12 VARIABLE
TCPCB      000032      INTEGER12 VARIABLE
TCMCD      000034      INTEGER12 VARIABLE
TCPD      000036      INTEGER12 VARIABLE
TCUDB      000040      INTEGER12 VARIABLE
TCPDB      000042      INTEGER12 VARIABLE

```

```

COMMON BLOCK /DATA/      LENGTH 000002

```



TC: 00000 INTEGER12 VARIABLE

FOR



```

0002 COMMON /DDB/ DDB
0009 COMMON /TC/ TCUM, TCRIM, TCRH, TCRG, TCUP, TCPS,
      TCUF, TCRF, TCUTP, TCRD, TCUD, TCRDC,

```

```

FORTRAN IV      U01C-03      MON 29-OCT-79 14:21:19      PAGE 002
CORE-08K, UIC=E1,123J      ,TT01=DK3:CONL05.FOR
*      TCUCB, TCRCB, TCUCD, TCRCD, TCUDB, TCRDB

```

```

0010 COMMON /DATAA/ TCC
      END OF DATA DEFINITION
      NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CHOPEN
      -----
      *****

```

```

      1) SET COMMAND
      2) CALL COMMON CODE FOR TRANSFER CODE 3 TYPE COMMANDS
      TCC=TCUCB
      IUCB=ITC3(1BUF,IADR)
      RETURN
      END

```

| FORTRAN IV | NAME  | OFFSET | STORAGE MAP | ATTRIBUTES |
|------------|-------|--------|-------------|------------|
| 00016      | ITC/T | 000016 | LOGICALX1   | ARRAY (8)  |
| 00026      | IUCB  | 000026 | INTEGERX2   | VARIABLE   |
| 00014      | IBUF  | 000014 | INTEGERX2   | PARAMETER  |
| 00000      | ITC3  | 000000 | INTEGERX2   | PROCEDURE  |
| 000100     | IADP  | 000100 | INTEGERX2   | VARIABLE   |

```

COMMON BLOCK /DDB/      LENGTH 000500
DDE  000000  INTEGER12 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/      LENGTH 000044
TCUIM 000000  INTEGER12 VARIABLE
TCRIM 000002  INTEGER12 VARIABLE
TCUGH 000004  INTEGER12 VARIABLE
TCURH 000006  INTEGER12 VARIABLE
TCURP 000010  INTEGER12 VARIABLE
TCURP 000012  INTEGER12 VARIABLE
TCURP 000014  INTEGER12 VARIABLE
TCURP 000016  INTEGER12 VARIABLE
TCURP 000020  INTEGER12 VARIABLE
TCURP 000022  INTEGER12 VARIABLE
TCURP 000024  INTEGER12 VARIABLE
TCURP 000026  INTEGER12 VARIABLE
TCURP 000030  INTEGER12 VARIABLE
TCURP 000032  INTEGER12 VARIABLE
TCURP 000034  INTEGER12 VARIABLE
TCURP 000036  INTEGER12 VARIABLE
TCURP 000040  INTEGER12 VARIABLE
TCURP 000042  INTEGER12 VARIABLE

COMMON BLOCK /DATA/    LENGTH 000002
TCC  000000  INTEGER12 VARIABLE

```

FORTRAN IV      U01C-03      MON 29-OCT-79 14:21:42      PAGE 001  
 CORE=08K, UIC=C1.123      ,TT0=DK3:CONL05.FOR

0001      C      INTEGER FUNCTION IRC8(IBUF)

FUNCTIONAL DESCRIPTION:

THIS ROUTINE IS CALLED TO READ THE BLOCK OF TEXT  
 FROM THE IBUF IN THE CORTAL. THIS WOULD BE USED  
 WHEN THE CORTAL INTERRUPTED VIA THE 'SEND MESSAGE' REQUEST,  
 TO READ THE TEXT TO DETERMINE WHAT THE MESSAGE WAS.

PARAMETERS:

IBUF = USER IBUF FOR TEXT

ERRORS:



0013 END

```
FORTRAN I/O STORAGE MAP
NAME OFFSET ATTRIBUTES
IRCB 000016 INTEGER*2 VARIABLE
ISUF 000014 INTEGER*2 PARAMETER VARIABLE
ITC3 000000 INTEGER*2 PROCEDURE
IADR 000070 INTEGER*2 VARIABLE

COMMON BLOCK /DDB/ LENGTH 000500
DDB 000000 INTEGER*2 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/ LENGTH 000044
TCUIM 000000 INTEGER*2 VARIABLE
TCRIM 000002 INTEGER*2 VARIABLE
TCUGM 000004 INTEGER*2 VARIABLE
TCURM 000006 INTEGER*2 VARIABLE
TCUPS 000010 INTEGER*2 VARIABLE
TCRPS 000012 INTEGER*2 VARIABLE
TCUFM 000014 INTEGER*2 VARIABLE
TCRFM 000016 INTEGER*2 VARIABLE
TCUTP 000020 INTEGER*2 VARIABLE
TCRTP 000022 INTEGER*2 VARIABLE
TCUDC 000024 INTEGER*2 VARIABLE
TCRDC 000026 INTEGER*2 VARIABLE
TCUCB 000030 INTEGER*2 VARIABLE
TCRCB 000032 INTEGER*2 VARIABLE
TCUCD 000034 INTEGER*2 VARIABLE
TCRCD 000036 INTEGER*2 VARIABLE
TCUDB 000040 INTEGER*2 VARIABLE
TCRDB 000042 INTEGER*2 VARIABLE

COMMON BLOCK /DATA/ LENGTH 000002
TCC 000000 INTEGER*2 VARIABLE
```

FORTRAN I/O U01C-03 MON 29-OCT-79 14:22:04 PAGE 001  
CORE=08K, UIC=C1,1231 ,TT01-DK3:COML05.FOR



FORTTRAN IV U01C-03 MON 29-OCT-79 14:22:04 PAGE 002  
 COME-ESK. UIC=C1,1233 ,TT01-DK31COML05.FOR

1) SET COMMAND  
 2) CALL COMMON CODE FOR TRANSFER CODE 3 TYPE COMMANDS

0010 C TCC-TCUDB  
 0011 C UDB-ITC3(1BUF,IADR)  
 0012 C RETURN  
 0013 C END

# FORTTRAN IV STORAGE MAP

| NAME | OFFSET | ATTRIBUTES                   |
|------|--------|------------------------------|
| UDB  | 000016 | INTEGER*2 VARIABLE           |
| 1BUF | 000014 | INTEGER*2 PARAMETER VARIABLE |
| ITC3 | 000000 | INTEGER*2 PROCEDURE          |
| IADR | 000070 | INTEGER*2 VARIABLE           |

COMMON BLOCK /DDB/ LENGTH 000500

DDB 000000 INTEGER\*2 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/ LENGTH 000044

|       |        |                    |
|-------|--------|--------------------|
| TCUIM | 000000 | INTEGER*2 VARIABLE |
| TCPIM | 000002 | INTEGER*2 VARIABLE |
| TCUGM | 000004 | INTEGER*2 VARIABLE |
| TCRGM | 000006 | INTEGER*2 VARIABLE |
| TCUPS | 000010 | INTEGER*2 VARIABLE |
| TCFPM | 000012 | INTEGER*2 VARIABLE |
| TCRFM | 000014 | INTEGER*2 VARIABLE |
| TCUTP | 000016 | INTEGER*2 VARIABLE |
| TCUTP | 000020 | INTEGER*2 VARIABLE |
| TCUDC | 000024 | INTEGER*2 VARIABLE |
| TCUDC | 000026 | INTEGER*2 VARIABLE |
| TCUCB | 000030 | INTEGER*2 VARIABLE |
| TCUCB | 000032 | INTEGER*2 VARIABLE |
| TCUCD | 000034 | INTEGER*2 VARIABLE |
| TCUCD | 000036 | INTEGER*2 VARIABLE |





```

      TCUCB, TCRCB, TCUCD, TCRCd, TCUDB, TCRD8
      COMMON /DATAA/ TCC
      END OF DATA DEFINITION
      NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CROPEN
      -----
      *****
      1) SET COMMAND
      2) CALL COMMON CODE FOR TRANSFER CODE 3 TYPE COMMANDS

```

```

      FORTRAN IU      U01C-03      MON 29-OCT-79 14:22:35      PAGE 002
      CORE-08K, UIC-E1.123J      ,TT0:-DK3:COML05.FOR

```

```

      C      TCC-TCRDB
      C      RDB-ITC3(1BUF, IADR)
      C      RETURN
      C      END

```

```

      FORTRAN IU      STORAGE MAP
      NAME      OFFSET      ATTRIBUTES
      RDB      000016      INTEGER12 VARIABLE
      IEUF      000014      INTEGER12 PARAMETER VARIABLE
      ITC3      000000      INTEGER12 PROCEDURE
      IADR      000070      INTEGER12 VARIABLE
      COMMON BLOCK /RDB/      LENGTH 000500
      DDB      000000      INTEGER12 ARRAY (8,20) VECTORED
      COMMON BLOCK /TC/      LENGTH 000044
      TCUM      000000      INTEGER12 VARIABLE
      TCPM      000002      INTEGER12 VARIABLE
      TCUM      000004      INTEGER12 VARIABLE

```

```

TCRJA 000006 INTEGER12 VARIABLE
TCRJB 000010 INTEGER12 VARIABLE
TCRJC 000012 INTEGER12 VARIABLE
TCRJD 000014 INTEGER12 VARIABLE
TCRJE 000016 INTEGER12 VARIABLE
TCRJE 000020 INTEGER12 VARIABLE
TCRJE 000022 INTEGER12 VARIABLE
TCRJE 000024 INTEGER12 VARIABLE
TCRJC 000026 INTEGER12 VARIABLE
TCRJC 000030 INTEGER12 VARIABLE
TCRJC 000032 INTEGER12 VARIABLE
TCRJC 000034 INTEGER12 VARIABLE
TCRJD 000036 INTEGER12 VARIABLE
TCRJD 000040 INTEGER12 VARIABLE
TCRJD 000042 INTEGER12 VARIABLE

```

```
COMMON BLOCK /DATA/ LENGTH 000002
```

```
TCC 000000 INTEGER12 VARIABLE
```

```

FORTRAN IV      V01C-03      MON 29-OCT-79 14:22:57      PAGE 001
CORE-08K, UIC-C1.1233      ,TT0-DK3:CONLOS.FOR

```

```
0001 C INTEGER FUNCTION IUCD(IBUF,IAD)
```

```
FUNCTIONAL DESCRIPTION:
```

THIS ROUTINE IS USED TO WRITE TO ANY BLOCK OF MEMORY IN THE CORTAL. ITS PRIMARY USE IS IN DOWN LOADING CODE TO THE CORTAL. HOWEVER ANY OF THE CORTAL MEMORY LOC. CAN BE READ, AND ANY OF THE RAM AREAS WRITTEN. THE ADDRESS MUST BE ON A 256 WORD BOUNDARY.

```
PARAMETERS:
```

```
IBUF - USER BUFFER OF TEXT
IAD - ADDRESS IN THE CORTAL FOR THE TRANSFER
      (MUST BE A 512 BYTE BOUNDARY)
```

```
ERRORS:
```

```
0 - SUCCESS
-1 - WRITE ERROR
```

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

```
DATA DEFINITION
```



```

PROGRAM I"  STORAGE MAP
NAME  OFFSET  ATTRIBUTES
IUCD  000020  INTEGER12 VARIABLE
IUCF  000014  INTEGER12 PARAMETER VARIABLE
IAD   000016  INTEGER12 PARAMETER VARIABLE
ITC3  000000  INTEGER12 PROCEDURE
IADR  000072  INTEGER12 VARIABLE

COMMON BLOCK /DDB/  LENGTH 000500
DDB  000000  INTEGER12 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/  LENGTH 000044
TCUIM  000000  INTEGER12 VARIABLE
TCRIM  000002  INTEGER12 VARIABLE
TCUGN  000004  INTEGER12 VARIABLE
TCRCM  000006  INTEGER12 VARIABLE
TCUPS  000010  INTEGER12 VARIABLE
TCRPS  000012  INTEGER12 VARIABLE
TCUFH  000014  INTEGER12 VARIABLE
TCRFH  000016  INTEGER12 VARIABLE
TCUTP  000020  INTEGER12 VARIABLE
TCRTP  000022  INTEGER12 VARIABLE
TCUDC  000024  INTEGER12 VARIABLE
TCRDC  000026  INTEGER12 VARIABLE
TCUCB  000030  INTEGER12 VARIABLE
TCRCB  000032  INTEGER12 VARIABLE
TCUCD  000034  INTEGER12 VARIABLE
TCRCD  000036  INTEGER12 VARIABLE
TCUDB  000040  INTEGER12 VARIABLE
TCRDB  000042  INTEGER12 VARIABLE

COMMON BLOCK /DATAA/  LENGTH 000002
TCC  000000  INTEGER12 VARIABLE

```

```

FORTRAN IU  U01C-03  MON 29-OCT-79 14:23:20  PAGE 001
CORE-08K, UIC-E1,123  ,TTO:-DK3:COML05.FOR

```

```

C 0001 C INTEGER FUNCTION IRCD(1BUF,1AD)
C C
C C FUNCTIONAL DESCRIPTION:
C C
C C

```

THIS ROUTINE IS USED TO READ FROM ANY BLOCK OF MEMORY  
IN THE CORTAL. THE ADDRESS MUST BE ON A 256 WORD  
BOUNDARY.

```

PARAMETERS:
  IBUF : USER BUFFER OF TEXT
  IAD  : ADDRESS IN THE COMITAL FOR THE TRANSFER
ERRORS:
  0 : SUCCESS
  -1 : WRITE ERROR

```

```

*****

```

```

DATA DEFINITION

```

```

INTEGER TCUIH, TCRIH, TCUGH, TORGH, TCUPS, TCRPS
INTEGER TCUFN, TCRFN, TCUTP, TCRTP, TCUDC, TCRDC
INTEGER TCUCB, TCRCB, TCUCD, TCRCD, TCUDB, TCRDB

```

```

INTEGER DDB(8,20)
INTEGER TCC

```

```

COMMON /DDB/ DDB

```

```

COMMON /TC/ TCUIH, TCRIH, TCUGH, TORGH, TCUPS, TCRPS,
             TCUFN, TCRFN, TCUTP, TCRTP, TCUDC, TCRDC,
             TCUCB, TCRCB, TCUCD, TCRCD, TCUDB, TCRDB

```

```

COMMON /DATAA/ TCC

```

```

END OF DATA DEFINITION

```

```

NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CMOPEN
-----

```

1) SET COMMAND  
2) CALL COMMON CODE FOR TRANSFER CODE 3 TYPE COMMANDS

```

0010 C TCC=TORCD
0011 C IRCD=ITC3(IBUF,IADR)
0012 C RETURN
0013 C END

```

# FORTRAN IV STORAGE MAP

| NAME | OFFSET | ATTRIBUTES                   |
|------|--------|------------------------------|
| IRCD | 000020 | INTEGER*2 VARIABLE           |
| IBUF | 000014 | INTEGER*2 PARAMETER VARIABLE |
| IAD  | 000016 | INTEGER*2 PARAMETER VARIABLE |
| ITC3 | 000000 | INTEGER*2 PROCEDURE          |
| IADR | 000072 | INTEGER*2 VARIABLE           |

| COMMON BLOCK /DB/ | LENGTH                          |
|-------------------|---------------------------------|
| DOB 000000        | INTEGER*2 ARRAY (8,20) VECTORED |

| COMMON BLOCK /TC/ | LENGTH             |
|-------------------|--------------------|
| TCUIM 000000      | INTEGER*2 VARIABLE |
| TCRIM 000002      | INTEGER*2 VARIABLE |
| TCUMH 000004      | INTEGER*2 VARIABLE |
| TCPCN 000006      | INTEGER*2 VARIABLE |
| TCUPS 000010      | INTEGER*2 VARIABLE |
| TCRPS 000012      | INTEGER*2 VARIABLE |
| TCUFH 000014      | INTEGER*2 VARIABLE |
| TCRFH 000016      | INTEGER*2 VARIABLE |
| TCUTP 000020      | INTEGER*2 VARIABLE |
| TCUTP 000022      | INTEGER*2 VARIABLE |
| TCUDC 000024      | INTEGER*2 VARIABLE |
| TCUDC 000026      | INTEGER*2 VARIABLE |
| TCUCB 000030      | INTEGER*2 VARIABLE |
| TCPCB 000032      | INTEGER*2 VARIABLE |
| TCUCD 000034      | INTEGER*2 VARIABLE |
| TCPCD 000036      | INTEGER*2 VARIABLE |
| TCUDB 000040      | INTEGER*2 VARIABLE |
| TCPCB 000042      | INTEGER*2 VARIABLE |

| COMMON BLOCK /DATA/ | LENGTH             |
|---------------------|--------------------|
| TCC 000000          | INTEGER*2 VARIABLE |





END OF DATA DEFINITION  
 NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION COPEN  
 -----  
 \*\*\*\*\*

FORTRAN IU U01C-03 MON 29-OCT-79 14:23:43 PAGE 002  
 CORE-08K, UIC-C1.123J ,TT0-DK3:COMLOS.FOR

```

0010 C IA=IAD
0011 C SET SUCCESS
0012 C ITC3=0
0013 C GET ADDRESS
0014 C DDB(4,TC)=IADDR(IBUF)
0015 C DDB(3,TC)=0
0016 C CHECK FOR READ/WRITE OF CODE
0017 C IF((TC.NE.TCUCD).AND.(TC.NE.TCRCD)) GO TO 100
0018 C FUNCTION = CODE DATA, SET TRANSFER COMMAND
0019 C FORCE TO EVEN 256 WORD PAGE
0020 C DDB(3,TC)=(IAD/256)*256
0021 C CALL HANDLER
0022 C IC=IWRITE(DDB(1,TC))
0023 C IF(IC.LT.0)ITC3=IC
0024 C RETURN
0025 C END
  
```

FORTRAN IU STORAGE MAP  
 NAME OFFSET ATTRIBUTES

```

IT:3 000020 INTEGER12 VARIABLE VARIABLE
IT:4 000014 INTEGER12 PARAMETER VARIABLE
IT:5 000016 INTEGER12 PARAMETER VARIABLE
IT:6 000072 INTEGER12 VARIABLE
IT:7 000000 INTEGER12 PROCEDURE
IT:8 000074 INTEGER12 VARIABLE
IT:9 000000 INTEGER12 PROCEDURE
IT:10 000000

```

```

COMMON BLOCK :DDB      LENGTH 000500

```

```

DDB  000000  INTEGER12 ARRAY (8,20) UECTORED

```

```

COMMON BLOCK :TC/      LENGTH 000044

```

```

TCUIM 000000  INTEGER12 VARIABLE
TCUIM 000002  INTEGER12 VARIABLE
TCUIM 000004  INTEGER12 VARIABLE
TCUIM 000006  INTEGER12 VARIABLE
TCUIM 000010  INTEGER12 VARIABLE
TCUIM 000012  INTEGER12 VARIABLE
TCUIM 000014  INTEGER12 VARIABLE
TCUIM 000016  INTEGER12 VARIABLE
TCUIM 000020  INTEGER12 VARIABLE
TCUIM 000022  INTEGER12 VARIABLE
TCUIM 000024  INTEGER12 VARIABLE
TCUIM 000026  INTEGER12 VARIABLE
TCUIM 000030  INTEGER12 VARIABLE
TCUIM 000032  INTEGER12 VARIABLE
TCUIM 000034  INTEGER12 VARIABLE
TCUIM 000036  INTEGER12 VARIABLE
TCUIM 000040  INTEGER12 VARIABLE
TCUIM 000042  INTEGER12 VARIABLE

```

```

COMMON BLOCK :DATA/    LENGTH 000002

```

```

TCC  000000  INTEGER12 VARIABLE

```

```

END

```

IURITU MACRO M110 29-OCT-79 14:26  
TABLE OF CONTENTS

2- 35 IADDR - MIMICK RT-11 SYSLIB FUNCTION IADDR  
3- 67 IURITU - MIMICK RT-11 IURITU FOR CORTAL RSX I/O

IURITU MACRO M110 29-OCT-79 14:26 PAGE 1

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30

FILE NAME: IURITU.MAC  
DATE LAST CHANGE: 12-APR-79  
18-SEP-79 JIM TINS  
INCORPORATE IODIG  
SHORTEN IURITU F:

HISTORY:  
DATE: MODIFICATION:  
11-APR-79 INITIAL WRITING  
12-APR-79 CHANGE TCD AND FCD

FUNCTION:  
THIS FILE CONTAINS TWO INTEGER FUNCTIONS CALLABLE  
FROM FORTRAN. THEY ARE:  
IURITU - MIMICKS THE IURITU SYSLIB FUNCTION IN RT-11.  
THIS ROUTINE WILL BE CALLED BY THE  
CORTAL FILE UTILITY (COMPIP).  
IADDR - THIS ROUTINE MIMICKS THE RT-11 FUNCTION IADDR

NOTE: BOTH OF THESE ROUTINES WILL BE  
CALLED BY RSX-11M PROGRAMS TO  
PERFORM RT-11 SYSLIB TYPE FUNCTIONS.

MACRO M1110 29-OCT-79 14:26 PAGE 2

```

33 .TITLE IURITU
34 .IDENT /SMCC/
35 .SBTTL IADDR - MIMICK RT-11 SYSLIB FUNCTION IADDR
36
37
38
39
40 .PCALL QIOU8S
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

```

IURITU MACRO M110 29-OCT-79 14:26 PAGE 3  
 - MIMICK RT-11 IURITU FOR CONTAL RSX I/O

67  
68

```

00000000
00000002
00000004
00000006
00000010
00000012
00000013
00000066 005725
00000010 012500
121

```

```

00000000
00000002
00000004
00000006
00000010
00000012
00000013
00000066 005725
00000010 012500
121

```

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121

```

INTEGER FUNCTION IURITU(BLOCKSIZ,DBDTABLE,BLOCKNUMBER,CHANNEL)  
 THIS FUNCTION CAN BE USED ONLY FOR CALLS TO THE CORTAL RSX-11M  
 HANDLER. THE FUNCTION MIMICS THE RT-11 SYSLIB  
 FUNCTION 'IURITU'. IT IS BEING WRITTEN  
 SO THAT THE CORTAL LIBRARY FOR RT-11 WILL NOT  
 REQUIRE MUCH MODIFICATION.  
 THE CORTAL LIBRARY HAS SEVERAL CALLS TO SYSLIB FUNCTION 'IURITU'.  
 THESE CALLS GENERATE REQUESTS TO THE RT-11 CORTAL HANDLER.  
 THIS ROUTINE WILL TAKE THE DATA FROM THE IURITU CALL AND MODIFY IT  
 TO A CALL TO QIO TO THE RSX-11M CORTAL HANDLER.  
 NOTE: THIS IS NOT A GENERAL PURPOSE IURITU TO QIO CONVERSION ROUTINE.  
 IT IS TAILORED TO THE PARAMETERS USED IN THE RT-11 CORTAL HANDLER.

BLOCKSIZ - IGNORED  
 DBDTABLE - TABLE OF CORTAL PARAMETERS DEFINED AS:

| OFFSET: | PARAMETER: | DEFINITION:                |
|---------|------------|----------------------------|
| 0       | CL(MASK)   | VALUE TO MASK CLEAR TRANS  |
| 2       | SET(MASK)  | MASK TO SET IN TRANSFER    |
| 4       | TC         | TRANSFER COMMAND           |
| 6       | BSA        | USER BUFFER ADDRESS        |
| 10      | BUC        | WORD COUNT                 |
| 12      | TCD        | TRANSFER CODE (BYTE)       |
| 13      | FCD        | READ WRITE FUNCTION (BYTE) |
|         |            | (3 = WRITE, 5 = READ)      |

NOTE: TCD AND FCD APPEAR TO HAVE THEIR DEFINITIONS REVERSED  
 THE DOS HANDLER FROM CORTAL DEFINES  
 THESE VARIABLES AS ABOVE. HOWEVER, THEY USE THEM IN THE  
 HANDLER) AS THOUGH THEIR ROLES  
 WERE REVERSED. - RK 12-APR-79

.GLOBL IURITU

```

CL = 0
SET = 2
TC = 4
BSA = 6
BUC = 10
TCD = 12
FCD = 13

```

```

IURITU: TST (RS)+
MOV (RS)+,R0
; IGNOR NUMBER OF PARAMS
; GET ADDRESS OF DBDTABLE

```

MACRO M1110 29-OCT-79 14:26 PAGE 3-1  
- NIMICK RT-11 IURITU FOR COMTAL RSX I/O

[illegible]

```

159 000160
160 000160
161 000236 000427
162 000236 000427
163 000240
164 000240
165 000316
166 000316
167 000316
168 000316
169 000364 015700 000036
170 000364 015700 000036
171 000364 015700 000036
172 000364 015700 000036
173 000364 015700 000036
174 000364 015700 000036
175 000364 015700 000036
176 000364 015700 000036
177 000364 015700 000036
178 000364 015700 000036
179 000364 015700 000036
180 000364 015700 000036

```

MACRO M1110 29-OCT-79 14:26 PAGE 3-2  
MIMICK RT-11 IURITU FOR COMTEL RSX 1/0

```

181 000370 012603
182 000372 012602
183 000374 012601
184 000376 000007
185 000376 000007
186 000376 000007
187 000376 000007
188 000376 000007
189 000376 000007
190 000376 000007
191 000376 000007
192 000376 000007
193 000376 000007
194 000376 000007
195 000376 000007
196 000376 000007
197 000376 000007
198 000376 000007
199 000376 000007
200 000376 000007
201 000376 000007
202 000376 000007
203 000376 000007
204 000376 000007
205 000376 000007
206 000376 000007
207 000376 000007
208 000376 000007
209 000376 000007
210 000376 000007
211 000376 000007
212 000376 000007
213 000376 000007
214 000376 000007
215 000376 000007

```

|                                          |           |       |          |
|------------------------------------------|-----------|-------|----------|
| MACRO M110                               | 29-OCT-79 | 14:26 | PAGE 3-3 |
| - MIMICK RT-11 IURITU FOR COMTAL RSX 1-0 |           |       |          |
| IURITU                                   | 000003    | 125   | 124      |
|                                          | 000006    | 106   | 040      |
|                                          | 000011    | 103   | 116      |
|                                          | 000014    | 117   | 111      |
|                                          | 000017    | 116   | 040      |
|                                          | 000022    | 102   | 115      |
|                                          | 000025    | 072   | 122      |
| 225                                      | 000000    | 105   | 000      |
| 226                                      | 000000    | 105   | 116      |
| 227                                      | 000003    | 117   | 122      |
|                                          | 000006    | 114   | 040      |
|                                          | 000011    | 040   | 111      |
|                                          | 000014    | 116   | 040      |
|                                          | 000017    | 117   | 040      |
|                                          | 000022    | 124   | 106      |
|                                          | 000025    | 124   | 105      |
|                                          | 000030    | 105   | 054      |
|                                          | 000033    | 040   | 040      |
|                                          | 000036    | 124   | 104      |
|                                          | 000041    | 043   | 111      |
|                                          | 000044    | 103   | 040      |
|                                          | 000047    | 000   | 050      |
| 228                                      |           |       | 051      |
| 229                                      |           |       |          |



230

000001

.END

IURITU MACRO M110 29-OCT-79 14:26 PAGE 3-4  
SYMBOL TABLE

FSH 000006  
ETCNT 000434P  
EUPADR 000432P  
BUC 000010  
CL 000000

DONE 000316R  
FCD 000013  
IADDR 000000RG  
IBYTEX 000400RG  
IUSTAT 000426R

IO.INL 002400  
IO.RPP 001240  
IO.TRM 002410  
IO.UPP 000640  
IS.SUC 000001

IURITU 000006RG  
READIT 000160P  
SET 000002  
TC 000004  
TCD 000012

TCMD 000440P  
TCODE 000432P  
UPIT 000240P  
SSSAPS 000002

ABS 000000  
000442  
001  
IGRAQU 000030  
002  
IMMQUE 000024  
003  
IFUNQU 000030  
004  
ICOMQU 000050  
005  
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 1208 WORDS ( 5 PAGES )  
DYNAMIC MEMORY: 3060 WORDS ( 11 PAGES )  
ELAPSED TIME: 00:01:34  
TTO:-DK3:IURITU.MAC  
MAC

### 3.0 COMTAL: MC<sup>2</sup> VERSION OF THE SRC SILHOUETTE PROGRAM

The objective of this documentation is to describe FORTRAN code written to convert the SRC "SILHOUETTE" program to produce its output on the COMTAL Vision One Image Display System. Two conversions were necessary to achieve this objective:

1. Conversion from the DEC RT-11 single user real-time system to the DEC RSX-11M multi-user real-time operating system.
2. Conversion of graphic output from the calligraphic VT-11 display device to the raster format COMTAL.

The first conversion was primarily a matter of replacing some RT-11 I/O conventions with RSX conventions. As a previous task in the contract under which this work was being performed involved integration of the COMTAL under PAR's "RIPS" system, the necessary I/O capabilities for the COMTAL under RSX were already available. Data for the SILHOUETTE program was transferred from RT-11 disks to Mc<sup>2</sup>'s RSX-11 disk by means of the DEC utility "FLX".

The aircraft files were copied directly in ASCII mode. It was discovered that the parameter and image files, when transferred in "images" mode, were not readable by the RSX version of the SILHOUETTE program. It was discovered that the RT-11 FORTRAN IV "unformatted read" statements used by SRC to create these files were incompatible with the RSX-11M FORTRAN IV "unformatted read" statements. After some experimenting, it was determined that if the "unformatted write" statements in the SRC RT-11 version were replaced with "unformatted direct access write" statements, then RSX-11M would be able to read them.

A conversion program, called BOBTST.FOR was written to translate both image and parameter files from RT-11 unformatted to RT-11 direct access (unformatted). The corresponding read statements in the Mc<sup>2</sup>

version of "SILHOUETTE" were change to direct access reads. Then "FLX" was used (image Mode) to copy the output of "BOBTST" to the Mc<sup>2</sup> RSX-11M disk for use by "SILHOUETTE".

Conversion of the graphic output of the SILHOUETTE program to a form acceptable to the COMTAL was a much more difficult task. Some form of lineal to raster conversion package had to be constructed to allow display of the aircraft files. The image files were somewhat simpler, requiring only coordinate system reversal (because of the COMTAL's implicit upper-left-hand-corner "origin"), translation, and 2-axis scaling, in addition to some linear expansion of each pixel's gradation. After some analysis, it was decided that because of the abbreviated time frame of the conversion effort, that program execution efficiency would be sacrificed, and that as few changes as possible would be made to the basic algorithms contained in the SRC program. Instead, Mc<sup>2</sup> wrote several routines which performed, in a raster environment, what the identically-named routines contained in the DEC RT-11, VT-11 FORTRAN support library performed.

The routine names and calling sequences were kept identical. Some new subroutines were written to accomplish, in software, what the VT-11 accomplished in hardware, primarily dealing with image origin and scaling. Routines supporting communication with the COMTAL under RSX-11M were task-built into the new SILHOUETTE executable task.

### 3.1 Operation of the New SILHOUETTE Program

#### 3.1.1 Preparation for Execution

##### 3.1.1.1 Conversion of Data Files

At present, only the aircraft files LEARJT.DAT, F15A.DAT and T33A.DAT have been converted from RT-11 to RSX-11M. The following instructions

should be followed to convert other (aircraft, parameter, image) file groupings that may be of interest.

o Parameter and Image files

- Mount Mc<sup>2</sup> RT-11 system packs on RK-05 drive 0 and the RT-11 data source pack on drive 1. Boot RT-11.
- RUN DK0:BOBTST. Respond to prompt "PARAMETER FILE NAME" by typing the full specification of the RT-11 parameter input file: i.e., DK1:T33A1P.DAT.
- Respond to prompt "ENTER DESTINATION FILE NAME" by typing the same file name, preceded by a different disk prefix, in this case: DK0:T33A1P.DAT.
- Respond to "IMAGE FILE NAME" by typing the full specification of the image file name, i.e.: DK1:T33A1.DAT.
- Respond to "ENTER DESTINATION FILE NAME" by typing the same file name, preceded by a different disk prefix, i.e.: DK0:T33A1.DAT.

As it exists, the program terminates with a "STOP" after each file pair converted, and must be re-run for every conversion desired.

- o Aircraft files were written as ASCII card images, and hence do not need the conversion provided by BOBTST.FOR.
- o Transfer of files from RT-11 to RSX-11M: The RSX utility "FLX" is used to transfer files from an RT-11 disk to an RSX-11M disk. The following instructions apply.
  - Mount Mc<sup>2</sup> RSX-11 disk on drive 0; RT-11 disk containing converted files on any other drive.

- Boot RSX and: SET /UIC=[1,123]
- Invoke "FLX".
- For each file triple, copy from RT-11 disk "n" to the disk labeled MCC:RSX, mounted on drive "m" as in the following example:
  - Copy aircraft file: DKm:=DKn:T33A.DAT/RT
  - Copy parameter file: DKm:=DKn:T33A1P.DAT/RT/IM
  - Copy image file: DKm:=DKn:T33A1.DAT/RT/IM

### 3.1.2 Preparation for Execution of "SILHOUETTE" Program

#### 3.1.2.1 Load The COMTAL Handler

The software device handler for the COMTAL must be loaded after system bootstrap, as it is normally non-resident in the RSX nucleus. Once loaded, it will remain resident until removed, or until the system is re-booted.

MCR>LOA ZB:

#### 3.1.2.2 Install the "SILHOUETTE" Task

MCR>INS DKm:COMTAL where m is the drive number of the Mc<sup>2</sup>:RSX disk.

#### 3.1.2.3 Run the Task

MCR>RUN COMTAL

### 3.1.3 Operator Interaction with the SILHOUETTE Program

#### 3.1.3.1 Introduction

An attempt was made to retain as much of the SRC SILHOUETTE prompting mechanism as possible. Some areas of difference exist because of the nature of the COMTAL, and an important improvement was made to the SRC version, allowing user specification of aircraft attitude in terms of yaw, pitch and roll angles, rather than the previous slant range and cross range unit orthogonal vectors, which were non-intuitive, and impossible to calculate mentally except in very special cases.

The other changes involve user specification of the COMTAL image memory to contain the resultant radar return image, and of the COMTAL graphic memory, which will contain the aircraft silhouette. There are three image memories and four graphic memories in the current Vision One, but Mc<sup>2</sup> programs preclude reading from or writing to the fourth graphic memory.

Translation of the aircraft silhouette, usually as a means of having it overlay the radar return image, was accomplished dynamically on the VT-11 by means of lightpen instruction. Such capability is not possible on the COMTAL because of the raster graphic organization, so the tracking function is replaced with a translation function that allows the user to specify how far (in pixels) the user wishes the image moved in the X and Y screen coordinate system. This translation process may be iterated as many times as desired, allowing at least the precision of the VT-11 lightpen translation.

### 3.1.3.2 User Interaction

```
SET UIC-E1,1233  
COM DE:  
INS DK3:CONTAL  
RUN CONTAL
```

AIRCRAFT SILHOUETTE DISPLAY  
-----

```
ENTER OPERATIONAL MODE ;  
1 - IMAGE DRIVEN MODE  
2 - STAND-ALONE MODE  
1  
SILHOUETTE FILE NAME  
DK3:F15A.DAT  
IMAGE FILE NAME  
DK3:F15A1.DAT  
PARAMETER FILE NAME  
DK3:F15A1P.DAT  
ENTER CONTAL IMAGE MEMORY NUMBER: (1-3)  
3  
ENTER CONTAL GRAPHIC MEMORY NUMBER (1-3):  
1  
TRANSLATE GRAPHIC (YES/NO)  
YES  
ENTER TRANSLATIONS: X,Y (REAL,SIGNED NUMBERS)  
-30.,-64.  
TRANSLATE GRAPHIC (YES/NO)  
NO  
NEW PROJECTION (YES/NO)  
NO  
NEW FILE (YES/NO)  
NO  
PROGRAM "SILHOUETTE" TERMINATED BY OPERATOR  
CONTAL -- STOP
```

Figure 3-1 Sample Command Printout

### 3.1.4 Building "COMTAL", The Mc<sup>2</sup> Version of "SILHOUETTE"

All source code, object modules, utilities and data for the "COMTAL" task reside on the disk labeled "MCC:RSX", under UIC=[1,123].

If modification to the adopted SILHOUETTE system is necessary, the user should perform the following steps:

- a. Edit and recompile the affected module(s)
- b. MCR>TKB @DK3:COMTAL

This second step invokes an indirect command file, which contains all information regarding the program contents and logical unit assignments of the resultant task "COMTAL".

### 3.2 Modification and Additions to SRC Software

#### 3.2.1 Modifications to SLMAIN.FTN

##### 3.2.1.1 Common Areas

GBUF - graphic airframe buffer and current X, current Y counters.  
LOWLFT- X, Y coordinates of VT-11 lower screen left, and X, Y translation quantities.  
LAST - X, Y coordinates of last point written to graphic image buffer.



### 3.2.1.2 Storage

- GBUF (8192) - 256 x 256 x 1 bit graphic pixel buffer, stored in screen rows, top to bottom.
- GBUF2 (32) - 256 x 1 bit graphic line buffer, for output routines.
- XSHFT, YSHFT- initially zero, these are displacements of the graphic image in COMTAL frame, created by the user's previous image translation requests. This allows translations from the last graphic image location, rather than from the initial position.

### 3.2.1.3 Logic Changes

To convert the SILHOUETTE Program to RSX-11M, it was necessary, as previously indicated, to translate the parameter and image files from RT-11 FORTRAN unformatted I/O to direct access unformatted I/O, which is compatible with RSX. It was also necessary to insert calls to a small subroutine, FLEASN (F I L E A S S I G N), that prompts the user to enter the names of the aircraft, parameter and image files. This function was previously performed by a special option within the RT-11 "ASSIGN" subroutine, which is not present in RSX FORTRAN. "Define File" Statements were added to supply information about the parameter and image files.

As an improvement to the original SRC method of specifying aircraft attitude in terms of unit slant range and cross range vectors, Mc<sup>2</sup> wrote a subroutine called "ROTATE" that accepts yaw, pitch and roll angles, and constructs the unit vectors from this information.

The VT-11 light pen tracking routine, which allowed the user to superimpose the aircraft image over the radar return image, has been replaced by a prompt requesting the user to enter, if he wishes to translate the image, the signed displacement of the aircraft image that is desired. This translation is measured in pixels and is referenced to the lower left corner of the screen. The COMTAL MACRO routine "INS" can be used to determine the translation values.

Variable SSZ, screen size in inches, was changed from 12. to 10.5, the visible dimension of the COMTAL.

#### 3.2.1.4 Known Problems

##### 3.2.1.4.1 Byte Exchange

The COMTAL supplied RSX-11 device handler does not swap bytes within a PDP-11 word during a transfer, as its RT-11 predecessor did. For this reason, routine IBYTEX is called to place the pixels in proper logical order before transfer.

##### 3.2.1.4.2 Specification of a New File While in Image-Driven Mode, (Mode 1)

Due to some quirk of FORTRAN, attempts to close a parameter and image file and re-open a new set of parameter and image files results in a file name specification error; that could not be fully understood because of missing RSX-11M FORTRAN documentation. As a temporary measure, the statement at line #47 will bypass this problem by directing all requests for "new file" while in Mode 1 to a graceful exit: the new file can be used by re-running "COMTAL".

### 3.2.2 Modification to SLSUBS.FTN

SLSUBS is a collection of FORTRAN subroutines, written by SRC, that has been modified as little as possible for use in the COMTAL version of SILHOUETTE. One routine, FLEASN, was added to read file names for the ASSIGN calls in SLMAIN.

The following COMMON areas and variables apply to one or more members of SLSUBS. Logic changes will be discussed on a routine-by-routine basis.

#### 3.2.2.1 Common Areas

- SCALE - holds scaling factor for aircraft SILHOUETTE construction.
- LAST - See SLMAIN.
- LOWLFT - See SLMAIN.
- GBUF - See SLMAIN.

#### 3.2.2.2 Storage

- GBUS (8192) - Graphic pixel buffer.
- COORD(6) - A temporary buffer for holding data values and applying the scale factor (SCL) to them.
- D(256) - Image line buffer, input (real data).
- ITEXT (8) - LOGICAL\*1 array to store "CLEAR IMAGE BUFFER" command.
- IMLINE - 256-byte image line output buffer.

#### 3.2.2.3 Logic Changes

##### 3.2.2.3.1 Subroutine Draw

Subroutine ADJUST, which multiplies all VT-11 coordinate positions by the scale factor SCL computed by Subroutine SCALE, appears directly before calls to routines writing points, lines and circles. This routine simulates the VT-11 hardware scaling feature.

#### 3.2.2.3.2 Subroutine CIRCLE

No changes.

#### 3.2.2.3.3 Subroutine CYLNDR

Other than call to ADJUST at Line #10, no changes.

#### 3.2.2.3.4 Subroutine PASS2

@ Lines #35-40, prompt for specification of image buffer to be used.

@ Lines #41-47, clears proper COMTAL image buffer.

@ Line #66, image intensity level is extended to 8 bits from the 3 bits of the VT-11.

@ Lines #72-75, write a line of the image to the COMTAL.

#### 3.2.2.3.5 Subroutine TVECT

No changes.

#### 3.2.2.3.6 Subroutine PLOT

No changes.

#### 3.2.2.3.7 Subroutine FLEASN

Allows user to enter filenames of aircraft, parameter and image files.

#### 3.2.3 COMGRF.FTN

COMGRF is a set of FORTRAN routines that emulate on the COMTAL routines of the same name in the DEC VT-11 support library. These routines allow most of the other SILHOUETTE software to be very similar to the RT-11 version.

#### 3.2.3.1 Subroutine SCAL (Min x, Min y, Max x, Max y)

Determines the spread between (Min x, Max x) and (Min y, Max y) and finds the larger difference, which is then divided by 1024 to get the VT-11 distance per unit increment. This is done because the COMTAL is a square display, rather than rectangular. The coordinates of the lower left corner of the display are determined by adding together 512 (to convert from the VT-11 SILHOUETTE convention of a center-screen origin) and four times the desired pixel displacement.

#### 3.2.3.2 Subroutine APNT

Resets the current x, current y counters to the x, y specified in the argument list. Currently unused by SILHOUETTE.

#### 3.2.3.3 Subroutine RDOT (DX, DY)

Moves the current x, y pointers to (CX+DX, CY+DY) and calls CDOT to illuminate that pixel.

#### 3.2.3.4 Subroutine LVECT (DX, DY)

LVECT is a lineal to raster conversion routine that "draws" a straight line in the graphic buffer between (CX, CY) and (CX+DX, CY+DY). LVECT determines the slope of this line. If the absolute value of the slope is less than 1, then the line is described by a formula of the form  $y = f(x)$ . Conversely, if the absolute value of the slope is  $>1$ , then the line is described in the form  $x=f(y)$ .

After the line's equation has been calculated, discrete values of the equations dependent variable are calculated by stepping the independent variable between its input limits in unit intervals. As the dependent variable is rarely of integer value, it is rounded up or down to its nearest integral value. Subroutine CDOT is called to place each (integral

independent variable, dependent variable) pair in the graphic display buffer. When all such pairs have been sent, the current x and current y counters are updated to reflect the end of the line.

#### 3.2.3.5 Subroutine CDOT (x, y)

Subroutine CDOT accepts pixel addresses in the VT-11 1024 x 1024 coordinates frame (Lower Left Origin), and converts these to pixels in the COMTAL 256 x 256 upper left origin frame. It then calculates the bit address of the pixel in array "GBUF" and sets this bit.

Quantities XLAST and YLAST contain the last x, y position set in the COMTAL buffer. If a request to set the same bit as set in the last call to CDOT (which often happens because of the 4:1 coordinate frame shrinkage), the request is ignored. Otherwise, XLAST, YLAST is reset to reflect the current request and the bits in GBUF are set.

#### 3.2.3.6 Subroutine ADJUST

This routine accepts an array of real variables and a counter N, and applies the multiplicative scale factor SCL to the first N values of the array. This allows one routine to suffice for line and point requests, which have three parameters requiring scaling, and the circle requests, which have four scalable parameters.

#### 3.2.4 Subroutine ROTATE

Subroutine ROTATE accepts three real, signed input parameters, which correspond to yaw, pitch and roll angles.

Yaw is measured clockwise if looking down onto top of aircraft. Pitch is measured positive if aircraft nose is pointed above horizontal, negative below. Roll is measured position clockwise when looking down the aircraft fuselage.

The input values, in signed degrees, are converted to radians, and unit vectors RS and RC are calculated from these values.

#### 3.2.5 VECTOR.FTN

The three vector algebra routines UNITV, DOTP and CROSSP are used without change from their RT-11 versions.

#### 3.2.6 Other Programs

##### 3.2.6.1 BOBTST.FTN

This stand-alone RT-11 program assists the user in copying RT-11 FORTRAN unformatted files to RT-11 direct-access format. The direct access format is compatible with RSX-11M FORTRAN, and can be "FLX'ed" from an RT-11 pack to an RSX-11M pack for use with the SILHOUETTE Program.

It is necessary to specify different disk packs for source and destination files when running this program so that the original file name can be retained. This is important because FLX will not support the renaming of files during transfer.

##### 3.2.6.2 COMTAL.CMD

This file is the task builder indirect command file that builds the task COMTAL, which is the Mc<sup>2</sup> version of the SRC task SILUET.SAV.

FORTRAN IV U01C-03 FRI 28-SEP-79 09:28:22 PAGE 001  
 CORE=08K, UIC=C1,1233 ,TT0:DK3:SLMAIN.FTN

AIRCRAFT SILHOUETTE DISPLAY PROGRAM  
 P.A.D.C. - PDP 11-40

MODIFIED BY ROBERT A. MCKIE, MEASUREMENT CONCEPT CORP.,  
 TO SUPPORT COMTAL VISION ONE  
 SEPTEMBER, 1979

PROGRAM FILES  
 -----

LOAD MODULE NAME                    • COMTAL.SAU

SOURCE FILES

MAIN PROGRAM                    • SLMAIN.FTN  
 DRAW,CIRCLE,CYLNDR,  
 PASS2,TUECT,TRACK,PLOT       • SLSUBS.FTN  
 UNITU,DOTP,CROSSP           • UECTOR.OBJ  
 SCAL,APMT,BDOT,LUECT,  
 CDOT,ADJUST                   • COMGRF.FTN  
 ROTATE                         • ROTATE.FTN

OBJECT FILES

MAIN PROGRAM                    • SLMAIN.OBJ  
 DRAW,CIRCLE,CYLNDR,  
 PASS2,TUECT,TRACK,PLOT       • SLSUBS.OBJ  
 UNITU,DOTP,CROSSP           • UECTOR.OBJ  
 GRAPHIC ROUTINES             • COMGRF.OBJ  
 YAU,PITCH,ROLL ENTRY         • ROTATE.OBJ

COMMON /GBUF/GBUF,CX,CY  
 COMMON /LOULFT,XLL,YLL,XSMT,YSMFT  
 COMMON /LAST,XLAST,YLAST  
 LOGICALS1 GBUF(8192)  
 LOGICALS1 GBUF2(32)  
 INTEGER AFNAME(7),INFILE(7),PARAF1(7)

0001  
 0002  
 0003  
 0004  
 0005  
 0006



```

0007 INTEGER AIRCFL,TTYI,TTYO
0008 REAL RS(3),RC(3),RS(3),FINUIS(2)
0009 INTEGER I2,VLAST,VLAST
0010 DATA AIRCFL,9,NING/10,NPRM,11/
0011 DATA TTYI/5,TTYO/7,NO/NO
0012 DATA DTR/.017453293/.552,10.5,SDT/1024./,EPS/.05/
0013 DATA NCRI/10/

```

```

FORTRAN IV      U01C-03      FRI 28-SEP-79 09:28:22      PAGE 002
CONE-08K, UIC-E1,123J      ,TT0=DK3:SLMAIN.PTN

```

```

0014 DATA DMIN/-100./,D8THR/-20./

```

# DEFINE FILES FOR PARAMETERS, IMAGE

## INITIAL INPUT PARAMETERS

```

MODE = OPERATIONAL MODE (1 OR 2)
NAME = DATA FILE NAME (10 CHARACTERS)
SCALE = SCALE FACTOR (METERS/INCH)
RS = SLANT RANGE VECTOR (3 VALUES)
RC = CROSS RANGE VECTOR (3 VALUES)

```

## CONSTRUCT AIRCRAFT SILHOUETTE

```

0015 WRITE(TTYO,29)
0016 WRITE(TTYO,26)
0017 READ(TTYI,12)MODE
0018 WRITE(TTYO,17)
0019 CALL FLEASN(TTYI,AFNAME)
0020 CALL ASSIGN(AIRCFL,AFNAME,0)
0021 IF (MODE.EQ.2) GO TO 42
0022 WRITE(TTYO,30)
0023 CALL FLEASN(TTYI,IMFILE)
0024 DEFINE FILE 10 (512,256,U,IAU)
0025 CALL ASSIGN(NING,IMFILE,0)
0026 WRITE(TTYO,31)
0027 CALL FLEASN(TTYI,PARAFL)
0028 DEFINE FILE 11 (1,256,U,IAU)
0029 CALL ASSIGN(NPRM,PARAFL,0)
0030 READ(NPRM,1)NUPDS,NRECS,RMAX,SRI,CRI,
0031 * (RS(1),I-1,3),(RC(1),I-1,3)
0032 GO TO 45
0033 40 DO 66 I2=1,8192

```



```

0070      1  CONTINUE
0071      IF(MODE.EQ.1) GO TO 45
0072      WRITE(TTYO,18)
0073      READ(TTYI,11)SCALE
0074      WRITE(TTYO,16)
0075      READ(TTYI,10)KTST
0076      IF(KTST.EQ.NO) GO TO 42
0077      CALL ROTATE(RS,RC)
0078      43  WRITE(TTYO,19)
0079      READ(TTYI,11)(RS(I),I=1,3)
0080      WRITE(TTYO,20)
0081      READ(TTYI,11)(RC(I),I=1,3)
0082      WRITE(TTYO,16)
0083      READ(TTYI,10)KTST
0084      IF(KTST.EQ.NO) GO TO 43
0085      CALL TUECT(RS,RC,RZ,ERS,KCMP)
0086      IF(KCMP.EQ.0) GO TO 48
0087      WRITE(TTYO,27)
0088      GO TO 43

```

```

PORTMAN IU      U01C-03      FRI 28-SEP-79 09:28:22      PAGE 004
COPE-08K, UIC-LI,123J      ,TT0:-DK3:SLMAIN.FTN
0095      48  CONTINUE
0096      IF(MODE.EQ.2) GO TO 59
0097      CALL PASS2(NING,DMIN,DBTHR,RIMX,
0098      * NURDS,NPECS,RMAX,SRI,CRI)

```

REQUEST USER TO ENTER GRAPHIC MEMORY NUMBER (1-3):

```

0099      59  WRITE(TTYO,106)
0100      READ(TTYI,12) IGMN
0101      IF(IGMN.GT.0.AND.IGMN.LT.4) GO TO 49
0102      WRITE(TTYO,107) IGMN
0103      GO TO 59
0104
0105      49  CALL DRAW(AIRCFL,RS,RC,RZ,SCALE,FINUIS,DTR,
0106      * NCPI,MODE,SSZ,SDT,RIMX,ISZ,IDIF,KCMP)

```

BYTE-SWAP ENTIRE BUFFER GBUF, DUE TO CORTAL RSX HANDLER

```

0036      CALL IBYTEX(GBUF,4096)
0037      NOW,WRITE GBUF TO THE CORTAL

```

```

0097 60 DO 700 NS-1,256
0098 JS = (NS-1)/32+1
0099 JS = (NS+32)
0100 DO 699 NS-15, JS
0101 GBUF2(NS-15+1) = GBUF(NS)
0102 699 CONTINUE
0103 CALL IUGM,IGMM,NS,GBUF2,
0104 700 CONTINUE
      DONE
      TRACKING OF UT-11 TO BE REPLACED BY A TEK 4014 REQUEST FOR
      IMAGE TRANSLATION X AND Y, EXPRESSED IN SIGNED PIXELS
      WRITE(TTYO,101)
0105 READ(TTYI,101) KTST
0106 IF (KTST.EQ.NO) GO TO 40
0107 WRITE(TTYO,102)
0108 READ(TTYI,102) XS,YS
0109 XSHT = XSHF + XS
0110 YSHF = YSHF + YS
0111 DO 55 IZ=1,8192
0112 55 GBUF(IZ) = 0
0113
0114

```

3-19

```

FORTPAN IU 0010-03 FRI 28-SEP-79 09:28:22 PAGE 005
COPE-02K, UIC=C1,123]
0115 C GO TO 49
0116 C 50 GO TO 40
      GRACEFUL EXIT ROUTINE
0117 999 WRITE(TTYO,108)
0118 STOP
      TERMINATION ROUTINE
0119 1000 WRITE(TTYO,105)
0120 STOP
0121 10 FORMAT(5A2)
0122 11 FORMAT(3F8.0)

```

```

0127 12 FORMAT('ARE YOU SURE - ')
0128 13 FORMAT('SILHOUETTE FILE NAME (')
0129 14 FORMAT('SCALE FACTOR (METERS/INCH) - ')
0130 15 FORMAT('SLANT RANGE VECTOR (3 VALUES) - ')
0131 16 FORMAT('CROSS RANGE VECTOR (3 VALUES) - ')
0132 17 FORMAT('NEW FILE (YES/NO) - ')
0133 18 FORMAT('NEW PROJECTION (YES/NO) - ')
0134 19 FORMAT('DISPLAY BUFFER INDEX - ')
0135 20 FORMAT('ENTER OPERATIONAL MODE - ')
0136 21 FORMAT('1 - IMAGE DRIVEN MODE - ')
0137 22 FORMAT('2 - STAND-ALONE MODE - ')
0138 23 FORMAT('CONDITION - ')
0139 24 FORMAT('RS AND RC MUST BE - ')
0140 25 FORMAT('ORTHOGONAL UNIT VECTORS - ')
0141 26 FORMAT('CONDITION - ')
0142 27 FORMAT('DISPLAY BUFFER (IBUF) - ')
0143 28 FORMAT('IS ABOUT TO OVERFLOW - ')
0144 29 FORMAT('AIRCRAFT SILHOUETTE DISPLAY - ')
0145 30 FORMAT('IMAGE FILE NAME - ')
0146 31 FORMAT('PARAMETER FILE NAME - ')
0147 32 FORMAT('214)
0148 101 FORMAT('TRANSLATE GRAPHIC (YES/NO) - ')
0149 102 FORMAT('ENTER TRANSLATIONS: X,Y (REAL,SIGNED NUMBERS) - ')
0150 103 FORMAT('2FS,3)
0151 104 FORMAT('PROGRAM "SILHOUETTE" TERMINATED BY OPERATOR - ')
0152 105 FORMAT('ENTER COMTAL GRAPHIC MEMORY NUMBER (1-3) - ')
0153 106 FORMAT('ENTER COMTAL IGMN ENTERED, "14" IS ILLEGAL, PLEASE RE-ENTER - ')
0154 107 FORMAT('PLEASE REUN "COMTAL" IF NEW FILES TO BE SPECIFIED - ')
0155 108 FORMAT('END

```

# CARTPHN I" STORAGE MAP

| NAME   | OFFSET | ATTRIBUTES           |
|--------|--------|----------------------|
| GEUF2  | 000006 | LOGICAL#1 ARRAY (32) |
| WNAME  | 000046 | INTEGER#2 ARRAY (7)  |
| INFILE | 000064 | INTEGER#2 ARRAY (7)  |
| PARPHL | 000102 | INTEGER#2 ARRAY (7)  |
| RS     | 000120 | REAL#4 ARRAY (3)     |
| RC     | 000134 | REAL#4 ARRAY (3)     |
| PL     | 000150 | REAL#4 ARRAY (3)     |
| FINUIS | 000164 | REAL#4 ARRAY (2)     |
| 4IDCFL | 000174 | INTEGER#2 VARIABLE   |
| TTVI   | 000202 | INTEGER#2 VARIABLE   |
| TTYO   | 000204 | INTEGER#2 VARIABLE   |
| NIMS   | 000176 | INTEGER#2 VARIABLE   |
| NPPM   | 000200 | INTEGER#2 VARIABLE   |
| NO     | 000206 | INTEGER#2 VARIABLE   |
| DTP    | 000210 | REAL#4 VARIABLE      |



```

COMMON BLOCK LOULT. LENGTH 000020
XL: 000000 REAL14 VARIABLE
YLL 000004 REAL14 VARIABLE
XSHFT 000010 REAL14 VARIABLE
VSHFT 000014 REAL14 VARIABLE

COMMON BLOCK LAST. LENGTH 000004
XLAST 000000 INTEGER12 VARIABLE
VLAST 000002 INTEGER12 VARIABLE
FOR

```





7 - R, PHI AND THETA FOR THE CIRCLE  
8 - TERMINATE DISPLAY

READ FILE, BRANCH ON FUNCTION CODE

FORTTRAN IU U01C-03 FRI 28-SEP-79 09:20:36 PAGE 002  
CORE-08K, UIC-1,1233 ,TT0:-DK3:SLSUBS.FTN

```

0011 IFLG = 0
0012 IRD = 1
0013 40 CONTINUE
0014 IF (IFLG.EQ.1) WRITE(7,808) IRD
0015 SOS FORMAT(' READ LINE 8, IS')
0016 READ(AIRCFL,10)NFC,(COORD(1),1-1.3)
0017 IRD = IRD + 1
0018 GO TO (41,42,43,44,45,46,47,48,49),NFC
0019 C
0020 41 CALL ADJUST(COORD,3)
0021 CALL DOTP(RC,COORD,X)
0022 CALL DOTP(RS,COORD,Y)
0023 CALL PLOT(X,Y,3)
0024 GO TO 50
0025 C
0026 42 CALL ADJUST(COORD,3)
0027 CALL DOTP(RC,COORD,X)
0028 CALL DOTP(RS,COORD,Y)
0029 CALL PLOT(X,Y,2)
0030 GO TO 50
0031 C
0032 43 GO TO 40
0033 C
0034 44 IF (MODE.EQ.2) GO TO 31
0035 DMUAL=100.*COORD(3)*RINX/(2.*COORD(2))
0036 GO TO 34
0037 31 IF (SCALE.NE.0) GO TO 32
0038 DMUAL=COORD(1)
0039 GO TO 34
0040 32 DMUAL=5528*COORD(2)*COORD(3)/(100.*SCALE)
0041 34 CALL SCAL7-DMUAL,-DMUAL,DMUAL,DMUAL)
0042 CALL PLOT(0.,0.,1)
0043 GO TO 50
0044 C
0045 45 NCIPC=COORD(1)
0046 CALL CYLNDP(AIRCFL,RS,RC,RZ,DTR,NCRI,NCIRC)
0047 GO TO 50
0048 C
0049 46 CONTINUE
0050 READ(AIRCFL,10)NFC,(COORD(1),1-4.6)

```

```

0045 CALL ADJUST(COORD,4)
0046 CALL CIRCLE(COORD,RS,RC,DTR,NCRI)
0047 GO TO 50
0048
0049 47 GO TO 40
0050
0051 48 CONTINUE
0052 GO TO 60
0053
0054 50 GO TO 40
0055
0056
0057
0058
0059
0060

```

FORTPAIN IU U01C-03 PAGE 003  
 CORE-08, UIC-E1,123J ,TTO:-DK3:SLSUBS.FTN

```

0055 60 CONTINUE
0056 REWIND AIRCFL
0057 RETURN
0058
0059 10 FORMAT(16,3F6.0)
0060 END

```

REWIND AND RETURN

| NAME   | OFFSET | STORAGE MAP | ATTRIBUTES          |
|--------|--------|-------------|---------------------|
| FINNIS | 000026 | REAL14      | PARAMETER ARRAY (2) |
| PC     | 000016 | REAL14      | PARAMETER ARRAY (3) |
| PC     | 000020 | REAL14      | PARAMETER ARRAY (3) |
| PC     | 000022 | REAL14      | PARAMETER ARRAY (3) |
| COMPD  | 000052 | REAL14      | PARAMETER ARRAY (3) |
| WIRCFI | 000014 | INTEGER12   | PARAMETER VARIABLE  |
| SCALE  | 000024 | REAL14      | PARAMETER VARIABLE  |
| DTP    | 000030 | REAL14      | PARAMETER VARIABLE  |
| NCPI   | 000032 | INTEGER12   | PARAMETER VARIABLE  |
| MODE   | 000034 | INTEGER12   | PARAMETER VARIABLE  |
| SSZ    | 000036 | REAL14      | PARAMETER VARIABLE  |
| SUT    | 000040 | REAL14      | PARAMETER VARIABLE  |

|        |        |          |           |          |
|--------|--------|----------|-----------|----------|
| PARAM  | 000042 | REAL14   | PARAMETER | VARIABLE |
| IF1    | 000044 | INTEGER2 | PARAMETER | VARIABLE |
| IF2    | 000046 | INTEGER2 | PARAMETER | VARIABLE |
| ACHF   | 000050 | INTEGER2 | PARAMETER | VARIABLE |
| IF0    | 000152 | INTEGER2 | PARAMETER | VARIABLE |
| WFC    | 000154 | INTEGER2 | PARAMETER | VARIABLE |
| I      | 000156 | INTEGER2 | PARAMETER | VARIABLE |
| ADJUST | 000000 | REAL14   | PROCEDURE |          |
| DOTP   | 000000 | REAL14   | PROCEDURE |          |
| x      | 000160 | REAL14   | VARIABLE  |          |
| y      | 000164 | REAL14   | VARIABLE  |          |
| PILOT  | 000000 | REAL14   | PROCEDURE |          |
| ORIGAL | 000170 | REAL14   | VARIABLE  |          |
| SCAL   | 000000 | REAL14   | PROCEDURE |          |
| NCIRC  | 000174 | INTEGER2 | VARIABLE  |          |
| CYNDP  | 000000 | REAL14   | PROCEDURE |          |
| SCALE  | 000000 | REAL14   | PROCEDURE |          |

COMMON BLOCK SCALE LENGTH 000004

| ST  | 202030 | REFLX4 | VARIABLE |
|-----|--------|--------|----------|
| 1   | 1      | 1      | 1        |
| 2   | 2      | 2      | 2        |
| 3   | 3      | 3      | 3        |
| 4   | 4      | 4      | 4        |
| 5   | 5      | 5      | 5        |
| 6   | 6      | 6      | 6        |
| 7   | 7      | 7      | 7        |
| 8   | 8      | 8      | 8        |
| 9   | 9      | 9      | 9        |
| 10  | 10     | 10     | 10       |
| 11  | 11     | 11     | 11       |
| 12  | 12     | 12     | 12       |
| 13  | 13     | 13     | 13       |
| 14  | 14     | 14     | 14       |
| 15  | 15     | 15     | 15       |
| 16  | 16     | 16     | 16       |
| 17  | 17     | 17     | 17       |
| 18  | 18     | 18     | 18       |
| 19  | 19     | 19     | 19       |
| 20  | 20     | 20     | 20       |
| 21  | 21     | 21     | 21       |
| 22  | 22     | 22     | 22       |
| 23  | 23     | 23     | 23       |
| 24  | 24     | 24     | 24       |
| 25  | 25     | 25     | 25       |
| 26  | 26     | 26     | 26       |
| 27  | 27     | 27     | 27       |
| 28  | 28     | 28     | 28       |
| 29  | 29     | 29     | 29       |
| 30  | 30     | 30     | 30       |
| 31  | 31     | 31     | 31       |
| 32  | 32     | 32     | 32       |
| 33  | 33     | 33     | 33       |
| 34  | 34     | 34     | 34       |
| 35  | 35     | 35     | 35       |
| 36  | 36     | 36     | 36       |
| 37  | 37     | 37     | 37       |
| 38  | 38     | 38     | 38       |
| 39  | 39     | 39     | 39       |
| 40  | 40     | 40     | 40       |
| 41  | 41     | 41     | 41       |
| 42  | 42     | 42     | 42       |
| 43  | 43     | 43     | 43       |
| 44  | 44     | 44     | 44       |
| 45  | 45     | 45     | 45       |
| 46  | 46     | 46     | 46       |
| 47  | 47     | 47     | 47       |
| 48  | 48     | 48     | 48       |
| 49  | 49     | 49     | 49       |
| 50  | 50     | 50     | 50       |
| 51  | 51     | 51     | 51       |
| 52  | 52     | 52     | 52       |
| 53  | 53     | 53     | 53       |
| 54  | 54     | 54     | 54       |
| 55  | 55     | 55     | 55       |
| 56  | 56     | 56     | 56       |
| 57  | 57     | 57     | 57       |
| 58  | 58     | 58     | 58       |
| 59  | 59     | 59     | 59       |
| 60  | 60     | 60     | 60       |
| 61  | 61     | 61     | 61       |
| 62  | 62     | 62     | 62       |
| 63  | 63     | 63     | 63       |
| 64  | 64     | 64     | 64       |
| 65  | 65     | 65     | 65       |
| 66  | 66     | 66     | 66       |
| 67  | 67     | 67     | 67       |
| 68  | 68     | 68     | 68       |
| 69  | 69     | 69     | 69       |
| 70  | 70     | 70     | 70       |
| 71  | 71     | 71     | 71       |
| 72  | 72     | 72     | 72       |
| 73  | 73     | 73     | 73       |
| 74  | 74     | 74     | 74       |
| 75  | 75     | 75     | 75       |
| 76  | 76     | 76     | 76       |
| 77  | 77     | 77     | 77       |
| 78  | 78     | 78     | 78       |
| 79  | 79     | 79     | 79       |
| 80  | 80     | 80     | 80       |
| 81  | 81     | 81     | 81       |
| 82  | 82     | 82     | 82       |
| 83  | 83     | 83     | 83       |
| 84  | 84     | 84     | 84       |
| 85  | 85     | 85     | 85       |
| 86  | 86     | 86     | 86       |
| 87  | 87     | 87     | 87       |
| 88  | 88     | 88     | 88       |
| 89  | 89     | 89     | 89       |
| 90  | 90     | 90     | 90       |
| 91  | 91     | 91     | 91       |
| 92  | 92     | 92     | 92       |
| 93  | 93     | 93     | 93       |
| 94  | 94     | 94     | 94       |
| 95  | 95     | 95     | 95       |
| 96  | 96     | 96     | 96       |
| 97  | 97     | 97     | 97       |
| 98  | 98     | 98     | 98       |
| 99  | 99     | 99     | 99       |
| 100 | 100    | 100    | 100      |

| COMMON | FLOCK  | LAST   | LENGTH | 000010 |
|--------|--------|--------|--------|--------|
| 000010 | 000010 | 000010 | 000010 | 000010 |

| 151 | 2200000 | REAL14 | VARIABLE |
|-----|---------|--------|----------|
|-----|---------|--------|----------|

33041 447434 402000 1571

**00000000**

| FILE   | 000004 | REAL14 | VARIABLE |
|--------|--------|--------|----------|
| 000004 | 000004 | 000004 | 000004   |

| NAME  | TYPE | LENGTH | POSITION | STATUS | DESCRIPTION |
|-------|------|--------|----------|--------|-------------|
| Y=HFT | REAL | 4      | 1        | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 5        | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 9        | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 13       | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 17       | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 21       | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 25       | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 29       | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 33       | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 37       | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 41       | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 45       | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 49       | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 53       | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 57       | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 61       | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 65       | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 69       | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 73       | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 77       | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 81       | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 85       | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 89       | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 93       | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 97       | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 101      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 105      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 109      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 113      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 117      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 121      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 125      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 129      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 133      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 137      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 141      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 145      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 149      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 153      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 157      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 161      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 165      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 169      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 173      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 177      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 181      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 185      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 189      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 193      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 197      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 201      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 205      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 209      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 213      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 217      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 221      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 225      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 229      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 233      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 237      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 241      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 245      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 249      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 253      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 257      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 261      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 265      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 269      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 273      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 277      | OK     | Y=HFT       |
| Y=HFT | REAL | 4      | 281      | OK     | Y=HFT       |
| Y=HFT |      |        |          |        |             |

```

MINI BLOCK : DEBUG /
LENGTH 000002

```

IFL'3 000000 INTEGER2 VARIABLE

COMMON BLOCK : GBUF / LENGTH 020010

ESTABLISHED IN 1953

DATE 10/10/10 135330 30/1/11

1 20181 40880 1210101 000000 31:27

|   |          |             |
|---|----------|-------------|
| ✓ | REAL I-4 | UNAVAILABLE |
| ✓ | REAL I-4 | UNAVAILABLE |
| ✓ | REAL I-4 | UNAVAILABLE |

FORTRAN IV UIC-03 FRI 28-SEP-79 09:21:15 PAGE 001  
CORE-03X, UIC-01.1233 ,TT01-DK3:SLSUBS.FTN

0001 SUBROUTINE CIRCLE (COORD,RS,RC,DTR,NCRI)  
0002 REAL COORD(6),RS(3),RC(3),TEMP(3)  
0003 EQUIVALENCE (X,TEMP(1)),(Y,TEMP(2)),(Z,TEMP(3))

PROJECT A CIRCLE  
CENTER (X0,Y0,Z0) - COORD(1,2 AND 3)  
RADIUS (R) - COORD(4)  
ORIENTATION (PHI,THETA) - COORD(5 AND 6)  
ONTO THE PLANE DEFINED BY THE  
SLANT AND CROSS RANGE VECTORS (RS,RC)

GET PARAMETERS  
POSITION BEAM TO 1ST POINT  
INITIALIZE SINE AND COSINE FUNCTIONS

X0=COORD(1)  
Y0=COORD(2)  
Z0=COORD(3)  
R=COORD(4)  
PHI=COORD(5)\*DTR  
THETA=COORD(6)\*DTR  
SPHI=SIN(PHI)  
CPHI=COS(PHI)  
STHETA=SIN(THETA)  
CTHETA=COS(THETA)  
X=RC\*PHI+X0  
Y=RS\*THETA+SPHI+Y0  
Z=-RS\*CTHETA+SPHI+Z0  
CALL DOTP(RC,TEMP,XP)  
CALL DOTP(RS,TEMP,YP)  
CALL PLOT(XP,YP,3)

SANG=SIN(DTR\*360./NCRI)  
CANG=COS(DTR\*360./NCRI)  
SNANG=0.  
CNANG=1.

DRAW 'NCRI' VECTORS AROUND THE CIRCLE

0024 N=NCRI+1  
0025 DO 44 I=1,N  
0026 XP=RC\*PHI  
0027 YP=RS\*THETA  
0028 Y=XP\*STHETA+SPHI+YP\*CTHETA+Y0  
0029 Z=-XP\*CTHETA+SPHI+YP\*STHETA+Z0  
0030

```

0031 CALL DOTP(RC,TEMP,XP)
0032 CALL DOTP(RS,TEMP,YP)
0033 CALL PLOT(XP,YP,2)
0034 XP=SNANG

```

FORTTRAN IU U01C-03 FRI 28-SEP-79 09:21:15 PAGE 002  
 CORE=0SA, UIC=C1,1233 ,TT0=DK3:SLSUBS.FTN

```

0035 VP=CNANG
0036 SNANG=XP*CNANG+YP*SNANG
0037 44 CNANG=VP*CNANG-XP*SNANG
      C
0038 RETURN
0039 END

```

| NAME   | OFFSET | STORAGE MAP | ATTRIBUTES          |
|--------|--------|-------------|---------------------|
| CODED  | 000014 | REAL14      | PARAMETER ARRAY (6) |
| PE     | 000016 | REAL14      | PARAMETER ARRAY (3) |
| TEMP   | 000020 | REAL14      | PARAMETER ARRAY (3) |
| DTP    | 000025 | REAL14      | ARRAY (3)           |
| NERI   | 000022 | REAL14      | PARAMETER VARIABLE  |
|        | 000024 | INTEGER12   | PARAMETER VARIABLE  |
|        | 000026 | REAL14      | PARAMETER VARIABLE  |
|        | 000032 | REAL14      | VARIABLE            |
|        | 000036 | REAL14      | VARIABLE            |
|        | 000046 | REAL14      | VARIABLE            |
|        | 000052 | REAL14      | VARIABLE            |
|        | 000055 | REAL14      | VARIABLE            |
|        | 000062 | REAL14      | VARIABLE            |
| PHI    | 000066 | REAL14      | VARIABLE            |
| THETA  | 000072 | REAL14      | VARIABLE            |
| SPHI   | 000076 | REAL14      | VARIABLE            |
| SIN    | 000080 | REAL14      | PROCEDURE           |
| CPHI   | 000102 | REAL14      | VARIABLE            |
| COS    | 000080 | REAL14      | PROCEDURE           |
| SMETH  | 000106 | REAL14      | VARIABLE            |
| CTHETA | 000112 | REAL14      | VARIABLE            |
| DOTP   | 000080 | REAL14      | PROCEDURE           |
| VP     | 000116 | REAL14      | VARIABLE            |
| YP     | 000122 | REAL14      | VARIABLE            |

```

PLOT          REAL14  PROCEDURE
S=0          REAL14  VARIABLE
C=0          REAL14  VARIABLE
S=0          REAL14  VARIABLE
C=0          REAL14  VARIABLE
N            INTEGER2 VARIABLE
I            INTEGER2 VARIABLE
000150

```

```

FORTRAN IV   UIC-03      FRI 28-SEP-79 09:21:49      PAGE 001
CORE-08K, UIC-E1.123J      ,T10:=DK3:SLSUBS.FTN

```

```

0001  SUBROUTINE CYLNDR (AIRCEL,RS,RC,RZ,DTR,NCRI,NCIRC)
0002  INTEGER AIRCEL
0003  REAL COORD(6),RS(3),RC(3),RZ(3)
0004  REAL ZC(3),S(3),T(3),X(2,10),Y(2,10)

```

```

      PROJECT A CYLINDER
      DEFINED AS A SET OF CONNECTED CIRCLES
      PROJECT THE CIRCLES AS USUAL
      CONSTRUCT THE LINE SEGMENTS TO BE CONNECTED
      DRAW THE LINE SEGMENTS

```

```

      NCR=MIN0(NCIRC,10)
      NDCR=0

```

```

      DO 48 N=1,NCR
      READ(AIRCEL,10)NFC,COORD(I),I=1,3)
      READ(AIRCEL,10)NFC,COORD(I),I=4,6)
      CALL ADJUST(COORD,4)
      CALL CIRCLE(COORD,RS,RC,DTR,NCRI)
      U=COORD(5)*DTR
      ZC(1)=SIN(U)
      CP=COS(U)

```

```

      U=COORD(6)*DTR
      ZC(2)=SIN(U)*CP
      ZC(3)=COS(U)*CP
      CALL CROSSP(RZ,ZC,T)
      CALL UNITUIT,T,U)
      IF(U.GE..00001) GO TO 42
      NDCP=NDCR+1
      GO TO 48

```

```

      42 DO 44 K=1,2
      L=2K-3
      DO 43 I=1,3
      S(I)=T(I)*COORD(4)*XL
      CALL DOTP(S,RC,X(K,N))
      CALL DOTP(COORD,PC,U)
      X(K,N)=X(K,N)+U
      CALL DOTP(S,RS,Y(K,N))
      CALL DOTP(COORD,RS,U)

```

```

0033      44 Y(K,N)=Y(K,N)+U
0034      48 CONTINUE
0035      MCR=MCR-NDCR
0036      IF(MCR.LT.2) RETURN
0037      DO 54 K=1,2
0038      CALL PLOT(X(K,1),Y(K,1),3)
0039      DO 54 N=2,MCR
0040      DO 54 CALL PLOT(X(K,N),Y(K,N),2)
0041      54 CONTINUE
0042      RETURN
0043      10 FORMAT(16,3F6.0)
0044      END

```

| NAME  | OFFSET | STORAGE MAP | ATTRIBUTES            |
|-------|--------|-------------|-----------------------|
| COOPD | 000032 | REAL14      | ARRAY (6)             |
| PS    | 000016 | REAL14      | PARAMETER             |
| PC    | 000020 | REAL14      | PARAMETER             |
| RC    | 000022 | REAL14      | PARAMETER             |
| SC    | 000026 | REAL14      | ARRAY (3)             |
| TC    | 000076 | REAL14      | ARRAY (3)             |
| UD    | 000076 | REAL14      | ARRAY (3)             |
| VC    | 000112 | REAL14      | ARRAY (2,10) VECTORED |
| WD    | 000126 | REAL14      | ARRAY (2,10) VECTORED |
| XD    | 000246 | REAL14      | PARAMETER             |
| YD    | 000014 | INTEGER12   | PARAMETER             |
| ZD    | 000024 | REAL14      | PARAMETER             |
| AD    | 000026 | INTEGER12   | PARAMETER             |
| BD    | 000030 | INTEGER12   | PARAMETER             |
| CD    | 000436 | INTEGER12   | PARAMETER             |
| DD    | 000000 | INTEGER12   | PROCEDURE             |
| ED    | 000040 | INTEGER12   | PROCEDURE             |
| FD    | 000442 | INTEGER12   | VARIABLE              |
| GD    | 000444 | INTEGER12   | VARIABLE              |
| HD    | 000446 | INTEGER12   | VARIABLE              |
| ID    | 000000 | REAL14      | PROCEDURE             |
| JD    | 000000 | REAL14      | PROCEDURE             |
| KD    | 000450 | REAL14      | VARIABLE              |
| LD    | 000000 | REAL14      | PROCEDURE             |
| MD    | 000454 | REAL14      | VARIABLE              |
| ND    | 000000 | REAL14      | PROCEDURE             |
| OD    | 000000 | REAL14      | PROCEDURE             |
| PD    | 000000 | REAL14      | PROCEDURE             |
| QD    | 000460 | INTEGER12   | VARIABLE              |
| RD    | 000462 | INTEGER12   | VARIABLE              |
| SD    | 000000 | REAL14      | PROCEDURE             |
| TD    | 000000 | REAL14      | PROCEDURE             |

FRI 28-SEP-79 09:22:07 PAGE 001  
F00TPM IN U01C-03  
CORE-05A, UIC-E, 1233 , TIO:DK3:SLSUBS.FTN

FOURPM 11 U01C-03  
CORE-USA, UIC-L:1237

```
00001 SUBROUTINE PASS2 (NINC,DIM,DTHR,RINX,  
00002 NURDS,NRECS,RPHAX,SRI,CRI )  
00003  
00004 REAL D(256)  
00005 LOGICAL IMLINE(256)  
00006 LOGICAL ITEXT(8)  
00007 INTEGER TYO,TYI  
00008 DATA TYO,TYI / 5,  
00009 DATA ITEXT / C,L, , , , , , , ,
```

# DISPLAY RADAR IMAGE RETURN

DETERMINE SLANT AND CROSS RANGE SCALE FACTORS  
 READ IMAGE RECORDS  
 CONVERT TO DB  
 CALCULATE INTENSITY (1-256)  
 DISPLAY POINTS

```

00000000 CPD=NURDS$CRI
00000001 SPD=NRDCS$SRI
00000002 IF(CPD.GT.SRD) GO TO 32
00000003 SPMAX=NRECS
00000004 CPMAX=NURDS$SRD/CPD
00000005 PIMX=SRD
00000006 CPI=.5*(NURDS-CRMAX)
00000007 CR2=CPMAX+CPI
00000008 YSP = CR2-CRI
00000009 YSP = SPMAX
00000010 XINC = .256./YSP
00000011 YINC = .256./YSP
00000012 YMIN = .01
00000013 YMIN = 0.
00000014 GO TO 40
00000015 32 CPMAX=NURDS
00000016 SPMAX=NRECS$SRD/SPD
00000017 PIMX=CPD
00000018 SRI=.5*(NRECS-SPMAX)
00000019 SP2=SPMAX+SRI
00000020 YSP = CPMAX
00000021 YSP = SR2-SRI
00000022 XINC = .256./YSP
00000023 YINC = .256./YSP
00000024 YMIN = 0.
00000025 YMIN = SRI
00000026 40 UPITE(TTV0,100)
00000027 PEADITV(101),IMNO

```



```

0037 IF (IMNO.GT.0.AND.IMNO.LT.4) GO TO 75
0038 WRITE(TTYO,102) IMNO
0039 GO TO 40

```

PROGRAM IV U01C-03 FRI 28-SEP-79 09:22:07 PAGE 002  
CODE-08A, UIC-C1.1232 ,TO:DK3:SLSUBS.FM

CLEAR IMAGE BUFFER

```

0041 75 IF (IMNO.EQ.1) ITEXT(6) = 49
0042 IF (IMNO.EQ.2) ITEXT(6) = 50
0043 IF (IMNO.EQ.3) ITEXT(6) = 51
0044 CALL IUCB(ITEXT)

```

```

0045 DO 48 I=1,NRECS
0046 DO 60 J=1,1256
0047 ILINE(18) = 0
0048 60 PENDING I(DIK),K=1,MURDS)
0049 DMAX=DMIN
0050 DO 42 K=1,MURDS
0051 RN=DIK/DIMAX
0052 DIK=DMIN
0053 IF RN.LE.0 GO TO 42
0054 DIK=AMAX1(DI,20,ZALOG10(RN))
0055 DMAX=AMAX1(DI,K,DMAX)
0056 42 CONTINUE
0057 IF (DMAX.LE.DBTHR) GO TO 48
0058 DO 44 K=1,MURDS
0059 X=(DBTHR-DIK)/DBTHR
0060 INTNS=AMAX1(0.,X)*256.

```

```

0061 KKK = FLOAT(K)*XINC
0062 IF (KKK.LT.1.OR.KKK.GT.256) GO TO 44
0063 IMLINE(KKK) = INTNS
0064 44 CONTINUE

```

NOV, WRITE A LINE TO THE CORTAL  
INVERT IMAGE Y; SCALE Y

```

0065 ILL = 1257 - (FLOAT(I)*YINC)
0066 IF (ILL.LT.1.OR.ILL.GT.256) GO TO 48
0067 CALL IUIA(IMNO,ILL,IMLINE)
0068 48 CONTINUE

```

FORTRAN IV STORAGE MAP

| NAME     | OFFSET | ATTRIBUTES |
|----------|--------|------------|
| 00000000 | 0      | 00000000   |
| 00000001 | 1      | 00000001   |
| 00000002 | 2      | 00000002   |
| 00000003 | 3      | 00000003   |
| 00000004 | 4      | 00000004   |
| 00000005 | 5      | 00000005   |
| 00000006 | 6      | 00000006   |
| 00000007 | 7      | 00000007   |
| 00000008 | 8      | 00000008   |
| 00000009 | 9      | 00000009   |
| 0000000A | 10     | 0000000A   |
| 0000000B | 11     | 0000000B   |
| 0000000C | 12     | 0000000C   |
| 0000000D | 13     | 0000000D   |
| 0000000E | 14     | 0000000E   |
| 0000000F | 15     | 0000000F   |
| 00000010 | 16     | 00000010   |
| 00000011 | 17     | 00000011   |
| 00000012 | 18     | 00000012   |
| 00000013 | 19     | 00000013   |
| 00000014 | 20     | 00000014   |
| 00000015 | 21     | 00000015   |
| 00000016 | 22     | 00000016   |
| 00000017 | 23     | 00000017   |
| 00000018 | 24     | 00000018   |
| 00000019 | 25     | 00000019   |
| 0000001A | 26     | 0000001A   |
| 0000001B | 27     | 0000001B   |
| 0000001C | 28     | 0000001C   |
| 0000001D | 29     | 0000001D   |
| 0000001E | 30     | 0000001E   |
| 0000001F | 31     | 0000001F   |
| 00000020 | 32     | 00000020   |
| 00000021 | 33     | 00000021   |
| 00000022 | 34     | 00000022   |
| 00000023 | 35     | 00000023   |
| 00000024 | 36     | 00000024   |
| 00000025 | 37     | 00000025   |
| 00000026 | 38     | 00000026   |
| 00000027 | 39     | 00000027   |
| 00000028 | 40     | 00000028   |
| 00000029 | 41     | 00000029   |
| 0000002A | 42     | 0000002A   |
| 0000002B | 43     | 0000002B   |
| 0000002C | 44     | 0000002C   |
| 0000002D | 45     | 0000002D   |
| 0000002E | 46     | 0000002E   |
| 0000002F | 47     | 0000002F   |
| 00000030 | 48     | 00000030   |
| 00000031 | 49     | 00000031   |
| 00000032 | 50     | 00000032   |
| 00000033 | 51     | 00000033   |
| 00000034 | 52     | 00000034   |
| 00000035 | 53     | 00000035   |
| 00000036 | 54     | 00000036   |
| 00000037 | 55     | 00000037   |
| 00000038 | 56     | 00000038   |
| 00000039 | 57     | 00000039   |
| 0000003A | 58     | 0000003A   |
| 0000003B | 59     | 0000003B   |
| 0000003C | 60     | 0000003C   |
| 0000003D | 61     | 0000003D   |
| 0000003E | 62     | 0000003E   |
| 0000003F | 63     | 0000003F   |
| 00000040 | 64     | 00000040   |
| 00000041 | 65     | 00000041   |
| 00000042 | 66     | 00000042   |
| 00000043 | 67     | 00000043   |
| 00000044 | 68     | 00000044   |
| 00000045 | 69     | 00000045   |
| 00000046 | 70     | 00000046   |
| 00000047 | 71     | 00000047   |
| 00000048 | 72     | 00000048   |
| 00000049 | 73     | 00000049   |
| 0000004A | 74     | 0000004A   |
| 0000004B | 75     | 0000004B   |
| 0000004C | 76     | 0000004C   |
| 0000004D | 77     | 0000004D   |
| 0000004E | 78     | 0000004E   |
| 0000004F | 79     | 0000004F   |
| 00000050 | 80     | 00000050   |
| 00000051 | 81     | 00000051   |
| 00000052 | 82     | 00000052   |
| 00000053 | 83     | 00000053   |
| 00000054 | 84     | 00000054   |
| 00000055 | 85     | 00000055   |
| 00000056 | 86     | 00000056   |
| 00000057 | 87     | 00000057   |
| 00000058 | 88     | 00000058   |
| 00000059 | 89     | 00000059   |
| 0000005A | 90     | 0000005A   |
| 0000005B | 91     | 0000005B   |
| 0000005C | 92     | 0000005C   |
| 0000005D | 93     | 0000005D   |
| 0000005E |        |            |

|         |           |             |
|---------|-----------|-------------|
| 000036  | REAL14    | ARRAY (256) |
| 002030  | LOGICAL1  | ARRAY (256) |
| 0020436 | LOGICAL1  | ARRAY (8)   |
| 000014  | INTEGER2  | VARIABLE    |
| 000016  | REAL14    | PARAMETER   |
| 000020  | REAL14    | PARAMETER   |
| 000022  | REAL14    | PARAMETER   |
| 000024  | REAL14    | PARAMETER   |
| 000026  | INTEGER2  | VARIABLE    |
| 000028  | INTEGER2  | VARIABLE    |
| 000030  | REAL14    | PARAMETER   |
| 000032  | REAL14    | PARAMETER   |
| 000034  | REAL14    | PARAMETER   |
| 0020478 | INTEGER32 | VARIABLE    |
| 0020479 | INTEGER32 | VARIABLE    |
| 0020480 | REAL14    | VARIABLE    |
| 002052  | REAL14    | VARIABLE    |
| 002056  | REAL14    | VARIABLE    |
| 002062  | REAL14    | VARIABLE    |
| 002066  | REAL14    | VARIABLE    |
| 002068  | REAL14    | VARIABLE    |
| 002070  | REAL14    | VARIABLE    |
| 002072  | REAL14    | VARIABLE    |
| 002074  | REAL14    | VARIABLE    |
| 002076  | INTEGER2  | VARIABLE    |
| 000000  | INTEGER2  | PROCEDURE   |
| 002710  | INTEGER2  | VARIABLE    |
| 002712  | INTEGER2  | VARIABLE    |
| 002714  | INTEGER2  | VARIABLE    |
| 002716  | REAL14    | VARIABLE    |
| 002722  | REAL14    | VARIABLE    |
| 000000  | REAL14    | PROCEDURE   |
| 000000  | REAL14    | PROCEDURE   |
| 002726  | REAL14    | VARIABLE    |
| 002732  | INTEGER2  | VARIABLE    |
| 002734  | INTEGER2  | VARIABLE    |

3-34

FOURTRAN II 001C-03 FRI 28-SEP-79 09:23:04 PAGE 001  
TT01=DK3:SLSUBS.FIN

```
0001 SUBROUTINE PLOT (X,Y,N)  
0002 COMMON /GBUF, GBUF, CX, CY  
0003 LOGICAL*1 GBUF(8192)
```

ILLUMINATE ABSOLUTE POINT COORDINATE  
FROM PREVIOUS BEAM POSITION

```

C
0004      GO TO (48,43,44),N
C
0005      43 CALL LUECT(X-CX,Y-CY)
0006      GO TO 48
C
0007      44 CALL RDOT(X-CX,Y-CY)
C
0008      48 CX=X
0009      CY=Y
C
0010      50 RETURN
0011      END

```

| FORTRAN IV           |        | STORAGE MAP   |                    |
|----------------------|--------|---------------|--------------------|
| NAME                 | OFFSET | ATTRIBUTES    |                    |
| /                    | 000014 | REAL14        | PARAMETER VARIABLE |
| /                    | 000016 | REAL14        | PARAMETER VARIABLE |
| N                    | 000020 | INTEGER12     | PARAMETER VARIABLE |
| LOCT                 | 000000 | INTEGER12     | PROCEDURE          |
| PDUT                 | 000000 | REAL14        | PROCEDURE          |
| COMMON BLOCK /CIBUF/ |        | LENGTH 020010 |                    |

```

GRUF 00000 LOGICAL*1 ARRAY (8192)
      02000 REAL*4 VARIABLE
      020004 REAL*4 VARIABLE

```

```

FOPTPAN IU 001C-03 FRI 28-SEP-79 09:23:17 PAGE 001
CODE=08K, UIC=C1,123J ,TT0=DK3:SLSUBS.FTN

```

00000

```

0001 SUBROUTINE FLEAS(LUN,FILNAM)
0002 INTEGER*2 FILNAM(7)
0003 PEAD(LUN,5)(FILNAM(1),I=1,7)
0004 5 FORMAT(7A2)
0005 RETURN
0006 END

```

```

FOPTPAN IU STORAGE MAP
NAME OFFSET ATTRIBUTES
FILNAM 000016 INTEGER*2 PARAMETER ARRAY (7)
LUN 000014 INTEGER*2 PARAMETER VARIABLE
I 000024 INTEGER*2 VARIABLE
F05.

```

FORTTRAN IU UIC-03 FRI 28-SEP-79 09:24:48 PAGE 001  
CORE-09X, UIC-01,1233 ,TT0:=DK3:CONGRF.FTN

```

*****
X
X THESE ROUTINES ARE A SUBSET OF THE
X UT-11 FORTTRAN SUPPORT ROUTINES CALLED
X BY THE SRC 'SILHOUETTE' PROGRAM, RE-WRITTEN
X FOR THE COMTAL VISION ONE.
X
X WRITTEN BY: ROBERT A MCKIE, SENIOR ASSOCIATE
X MEASUREMENT CONCEPT COMP.
X 9/79
X
*****

```

SUBROUTINE SCAL(X1,Y1,X2,Y2)

```

COMMON /SCALE/ SCL
COMMON /LOUFT/XLL,YLL,XSFT,YSFT
SUBROUTINE TO REPLACE UT-11 SCAL ROUTINE
TO COMPUTE A SCALE FACTOR FOR SUBSEQUENT DRAWINGS
ON THE COMTAL VISION ONE. UNIFORM SCALING IN BOTH X
AND Y DIRECTIONS IS COMPUTED, BASED ON WHETHER
THE DISTANCE BETWEEN X1 AND X2 IS SMALLER OR GREATER THAN THAT
BETWEEN Y1 AND Y2: THE GREATER DISTANCE IS USED TO DETERMINE
SCALE 'SCL'.

```

```

A=X2 - X1
B=Y2 - Y1
XLL = 512. + (4.*XSFT)
YLL = 512. + (4.*YSFT)
IF(ABS(A).LT.ABS(B)) GO TO 10
SCL = 1024./A
GO TO 20
10 SCL = 1024./B
20 CONTINUE
END

```

| FORTRAN IV           |        | STORAGE MAP   |                    |
|----------------------|--------|---------------|--------------------|
| NAME                 | OFFSET | ATTRIBUTES    |                    |
| X1                   | 000014 | REAL14        | PARAMETER VARIABLE |
| Y1                   | 000016 | REAL14        | PARAMETER VARIABLE |
| X2                   | 000020 | REAL14        | PARAMETER VARIABLE |
| Y2                   | 000022 | REAL14        | PARAMETER VARIABLE |
| A                    | 000024 | REAL14        | VARIABLE           |
| B                    | 000030 | REAL14        | VARIABLE           |
| ABS                  | 000000 | REAL14        | PROCEDURE          |
| COMMON BLOCK /SCALE/ |        | LENGTH 000004 |                    |
| SCL                  | 000000 | REAL14        | VARIABLE           |
| COMMON BLOCK /LOULFT |        | LENGTH 000020 |                    |
| XLL                  | 000000 | REAL14        | VARIABLE           |
| YLL                  | 000004 | REAL14        | VARIABLE           |
| XSHFT                | 000010 | REAL14        | VARIABLE           |
| YSHFT                | 000014 | REAL14        | VARIABLE           |

FORTRAN IV U01C-03 PAGE 001  
 FUSE-08K, UIC-11,123J ,TT0-DK3:COMGRF.FTN

0001 SUBROUTINE APNT(X,Y,IP1,IP2,IP3,IP4)  
 0002 CX = X  
 0003 CY = Y  
 0004 RETURN  
 0005 END

| FORTRAN IV |        | STORAGE MAP |                    |
|------------|--------|-------------|--------------------|
| NAME       | OFFSET | ATTRIBUTES  |                    |
| X          | 000014 | REALx4      | PARAMETER VARIABLE |
| Y          | 000016 | REALx4      | PARAMETER VARIABLE |
| IP1        | 000020 | INTEGERx2   | PARAMETER VARIABLE |
| IP2        | 000022 | INTEGERx2   | PARAMETER VARIABLE |
| IP3        | 000024 | INTEGERx2   | PARAMETER VARIABLE |
| IP4        | 000026 | INTEGERx2   | PARAMETER VARIABLE |
| CX         | 000030 | REALx4      | VARIABLE           |
| CY         | 000034 | REALx4      | VARIABLE           |

FORTRAN IV U010-03  
 CORE=08K, UIC=E1,1233  
 FRI 28-SEP-79 09:25:11  
 ,TT0=-DK3:CONGRF.FTN  
 PAGE 001

|      |   |  |                          |
|------|---|--|--------------------------|
| C    |   |  |                          |
| C    |   |  |                          |
| C    |   |  |                          |
| C    |   |  |                          |
| 0001 | C |  | SUBROUTINE RDOT(DX,DY)   |
| 0002 | C |  | COMMON /GBUF/ GBUF,CX,CY |
| 0003 | C |  | LOGICALx1 GBUF(8192)     |
| 0004 | C |  | IX=CX+DX                 |
| 0005 | C |  | IY=CY+DY                 |
| 0006 | C |  | CX=CX+DX                 |
| 0007 | C |  | CY=CY+DY                 |
| 0008 | C |  | CALL CDOT (IX,IY)        |
| 0009 |   |  | RETURN                   |
| 0010 |   |  | END                      |

| FORTRAN IV |        | STORAGE MAP |                    |
|------------|--------|-------------|--------------------|
| NAME       | OFFSET | ATTRIBUTES  |                    |
| DX         | 000014 | REALx4      | PARAMETER VARIABLE |
| DY         | 000016 | REALx4      | PARAMETER VARIABLE |



```

      IN      INTEGER*2 VARIABLE
      IN      INTEGER*2 VARIABLE
      IN      REAL*4   PROCEDURE

```

```

COMMON BLOCK  GBUF,      LENGTH 620010
GBUF          LOGICAL*81 ARRAY (8192)
CX            REAL*4   VARIABLE
CY            REAL*4   VARIABLE

```

```

FORTRAN IV      UIC-03      FRI 28-SEP-79 09:25:22      PAGE 001
CORE=08K, UIC=C1,123      ,TT0:=DK3:COMGRF.FTN

```

```

SUBROUTINE LUECT(DX,DY)

```

```

      SIMULATES UT-11 LUECT, WHICH DRAWS A CONTINUOUS LINE BETWEEN
      CURRENT CURSOR LOCATION, (X,Y), AND (X+DX,Y+DY). AS THE CORTAL GRAPHIC
      MEMORY IS A 256 X 256 DOT MATRIX, THIS ROUTINE WILL GENERATE
      THE NECESSARY DOTS TO APPROXIMATE THE INDICATED STRAIGHT LINE.
      PARAMETER 'T' IS USED TO ROUTE LUECT CALLS OF NEAR ZERO OR INFINITE
      SLOPE TO STMT 8 5,(V*F(X)), OR TO STMT 8 55 (X*G(Y)), OR TO RETURN
      WITHOUT DRAWING IF BOTH DX AND DY ARE VERY SMALL.

```

```

      COMMON /GBUF/ GBUF,CX,CY
      COMMON /DEBUG/ IFLG
      LOGICAL*8 GBUF(8192)
      REAL *8 T,MP,BP
      INTEGER Y1,Y2,X1,X2
      DATA T/.000005/

```

```

      COMPUTE: EQUATION OF LINE THROUGH (CX,CY) AND (CX+DX,CY+DY)

```

```

      CHECK FOR INFINITE SLOPE:

```

```

      IF(ABS(DX).LT.T.AND.ABS(DY).LT.T) GO TO 11
      IF(ABS(DX).LT.T) GO TO 55
      IF(ABS(DY).LT.T) GO TO 5
      IF(ABS(DY).GT.ABS(DX)) GO TO 55

```

THIS CODE FOR Y = F(X)

0015 S M = DY/DX  
0017 B = CY - MSCX

0018 DO 10 I=1, IABS(INT(DX))  
0019 X = CX + SIGN(FLOAT(I), DX)

0020 COMPUTE: \*TRUE (FLOATING POINT) Y\*

TV = MXX + B

0021 DECIDE WHICH IS "NEARER" PIXEL

FORTRAN IU U01C-03 PAGE 002  
CORE-08K, UIC-C1.123J FRI 28-SEP-79 09:25:22 ,TT01-DK3:COMGRF.FIN

0021 Y1 = INT(TV)  
0022 Y2 = Y1 + ISIGN(1, INT(DV))  
0023 IF (ABS(TV-FLOAT(Y1)).GT.ABS(TV-FLOAT(Y2))) GO TO 30

0025 20 IV=Y1  
0026 GO TO 50

0027 30 IV = Y2

THE CALL TO CDOT WILL PLACE AN ILLUMINATED POINT AT POSITION (X,Y)  
OF THE 256 X 256 COMTAL GRAPHIC MEMORY.

0028 50 IX=X  
0029 CALL CDOT(IX,IV)

0030 10 CONTINUE  
0031 CX=CX+DX  
0032 CY=CY+DY  
0033 RETURN

THIS CODE FOR X = F(Y)

```

0034 55 MP = DX/DY
0035 BP = CX-MP*CV
0036 DO 60 I=1,INT(DY))
0037 Y-CV = SIGN(FLOAT(I),DY)
0038 TX = MP*Y + BP
0039 X1 = INT(TX)
0040 X2 = X1 + ISIGN(1,INT(DX))
0041 IF(ABS(TX-FLOAT(X1)).GT.ABS(TX-FLOAT(X2))) GO TO 70
0042 IX = X1
0043 GO TO 75
0044 70 IX = X2
0045 75 IX = Y
0046 CALL CDOT TO ILLUMINATE PIXEL

```

```

0000 CALL CDOT TO ILLUMINATE PIXEL

```

```

0047 CALL CDOT(IX,IV)
0048 60 CONTINUE + DX
0049 CX = CX + DX
0050 CY = CY + DY
0051 RETURN

```

FRI 28-SEP-79 00:25:22 PAGE 003  
CORE-08X, UIC-L1.123J ,TTO:-DK3:CONCRF.FTN

FOPTPRN IU 001C-03  
CORE-08X, UIC-L1.123J

0052 END

| NAME | OFFSET | ATTRIBUTES | STORAGE MAP |
|------|--------|------------|-------------|
| DX   | 000014 | REAL34     | PARAMETER   |
| DY   | 000016 | REAL34     | PARAMETER   |
| MP   | 000026 | REAL34     | VARIABLE    |
| BT   | 000032 | REAL34     | VARIABLE    |
| BP   | 000020 | REAL34     | VARIABLE    |
| BP   | 000036 | REAL34     | VARIABLE    |
| Y1   | 000042 | REAL34     | VARIABLE    |
| Y2   | 000046 | REAL34     | VARIABLE    |
| X1   | 000050 | INTEGER32  | VARIABLE    |
| X1   | 000052 | INTEGER32  | VARIABLE    |

```

X2=
M5=
I=
INT=
X=
SIGN=
FLOAT=
TN=
ISIGN=
IV=
IN=
CDOT=
Y=
TN=
NN054 INTEGER12 VARIABLE
NN055 REAL14 PROCEDURE
NN056 INTEGER12 VARIABLE
NN057 INTEGER12 PROCEDURE
NN058 INTEGER12 PROCEDURE
NN059 INTEGER12 PROCEDURE
NN060 REAL14 PROCEDURE
NN061 REAL14 PROCEDURE
NN062 REAL14 PROCEDURE
NN063 REAL14 PROCEDURE
NN064 REAL14 PROCEDURE
NN065 REAL14 PROCEDURE
NN066 REAL14 PROCEDURE
NN067 REAL14 PROCEDURE
NN068 REAL14 PROCEDURE
NN069 REAL14 PROCEDURE
NN070 REAL14 PROCEDURE
NN071 REAL14 PROCEDURE
NN072 REAL14 PROCEDURE
NN073 REAL14 PROCEDURE
NN074 REAL14 PROCEDURE
NN075 REAL14 PROCEDURE
NN076 REAL14 PROCEDURE
NN077 REAL14 PROCEDURE
NN078 REAL14 PROCEDURE
NN079 REAL14 PROCEDURE
NN080 REAL14 PROCEDURE
NN081 REAL14 PROCEDURE
NN082 REAL14 PROCEDURE
NN083 REAL14 PROCEDURE
NN084 REAL14 PROCEDURE
NN085 REAL14 PROCEDURE
NN086 REAL14 PROCEDURE
NN087 REAL14 PROCEDURE
NN088 REAL14 PROCEDURE
NN089 REAL14 PROCEDURE
NN090 REAL14 PROCEDURE
NN091 REAL14 PROCEDURE
NN092 REAL14 PROCEDURE
NN093 REAL14 PROCEDURE
NN094 REAL14 PROCEDURE
NN095 REAL14 PROCEDURE
NN096 REAL14 PROCEDURE
NN097 REAL14 PROCEDURE
NN098 REAL14 PROCEDURE
NN099 REAL14 PROCEDURE
NN100 REAL14 PROCEDURE

```

```

COMMON BLOCK /GBUF/ LENGTH 020010
GBUF 000000 LOGICAL11 ARRAY (8192)
GBUF 020000 REAL14 VARIABLE
GBUF 020004 REAL14 VARIABLE

COMMON BLOCK /DEBUG/ LENGTH 000002
IFLG 000000 INTEGER12 VARIABLE

```

EUPTRAN IU U01C-03 PAGE 001  
 COPE-08K, UIC-01,1233 ,TT0:-DK3:CONGRP.FTM

```

0001 SUBROUTINE CDOT(X,Y)
      CDOT SETS A BIT IN ARRAY "GBUF" CORRESPONDING TO PIXEL
      LOCATION (X,Y). GBUF IS ORGANIZED AS CONSECUTIVE
      "LINES" OF BITS, WHERE 256 BITS 22 BYTES. THE LINES
      ARE STORED IN DESCENDING ORDER (256-1) SO THAT THE LOWER LEFT
      ORIGIN CONVENTION OF THE UT-11 IS PRESERVED.

```

```

0002 COMMON /GBUF/ GBUF,CX,CY
0003 COMMON /LOWFT/ XLL,YLL,XSHFT,YSHFT
0004 COMMON /DEBUG/ IFLG
0005 COMMON /LAST/ XLAST,YLAST
0006 LOGICAL11 GBUF(8192)
0007 LOGICAL11 BITS(8)

```

INTEGER2 UNINDEX,BINDEX,X,Y,XLAST,YLAST  
 DATA BITS 128,64,32,16,8,4,2,1  
 CONVERT TO CONTAL COORDINATE FRAME BY TRANSLATING LOWER LEFT CORNER  
 TO (0,0) AND REDUCING RASTER COUNT BY A FACTOR OF 4 (1024/4 = 256).

XT = (FLOAT(X)+XLL)/4.  
 YT = (FLOAT(Y)+YLL)/4.

X = (X+INT(XLL))/4  
 Y = (Y+INT(YLL))/4

DETERMINE 'NEAREST' PIXEL

IF((XT-FLOAT(X)).GT.(.5)) X = X+1  
 IF((YT-FLOAT(Y)).GT.(.5)) Y = Y+1

NOW IN CONTAL FRAME

CHECK TO SEE IF SAME AS LAST POINT PLOTTED  
 THIS SAVES A LITTLE TIME BY NOT REPEATING  
 THE STORAGE OF SEQUENTIALLY IDENTICAL X,Y'S.

IF(Y.EQ.XLAST.AND.Y.EQ.YLAST) GO TO 35  
 NOT SAME, SO RESET XLAST, YLAST

FORTPAN I/O U01C-03 PAGE 002  
 COPE-08K, UIC-E1,123 ,TT0:=DK3:CONGRF.FTM

XLAST = X  
 YLAST = Y

NOW, CHECK RANGE OF X AND Y. ARE THEY WITHIN 0-255?

IF(X.LT.0.OR.X.GT.255) GO TO 35  
 IF(Y.LT.0.OR.Y.GT.255) GO TO 35

OK - PROCEED

COMPUTE: INDEX TO GBUF CORRESPONDING TO X,Y

[illegible]

010020 HL3M37 / 5185 / 20078 W/100000

000000 000020  
LENGTH / 15079  
M5079 000000

| COMMON | ALYX  | DEB   | LENGTH | 00000 |
|--------|-------|-------|--------|-------|
| 00000  | 00000 | 00000 | 00000  | 00000 |

FRI 28-SEP-79 09:26:29  
 .:T0:=DK3:COMGRF.FTN  
 PAGE 001  
 U010-03  
 FORTRAN IV  
 CODE=08K, UIC=C1.123

SUBROUTINE ADJUST(A,N)

ALTERS THE PARAMETERS OF ALL CALLS TO "LJECCT", "RDOT", SO AS TO REFLECT THE RESULTS OF SUBROUTINE "SCAL". THE ORIGINAL DEC "J"-1 SUPPORT ROUTINE TOOK CARE OF THIS SCALING IN HARDWARE REGISTERS. THE COMTAL VERSION MUST PERFORM THIS FUNCTION IN SOFTWARE.

```

0002 COMMON /SCALE/ SCL
0003 DIMENSION A(6)
0004 DO 10 I=1,N
0005 A(I) = A(I)*SCL
0006 10 CONTINUE
0007 RETURN
0008 END

```

| NAME      | OFFSET | ATTRIBUTES | STORAGE MAP | FORTRAN I/O |
|-----------|--------|------------|-------------|-------------|
| REAL14    | 000014 | PARAMETER  | ARRAY (6)   |             |
| INTEGER12 | 000016 | PARAMETER  | VARIABLE    |             |
| INTEGER12 | 000020 | PARAMETER  | VARIABLE    |             |

COMMON BLOCK SCALE LENGTH 000004  
SCL 000000 REAL14 VARIABLE  
FOR



FRI 28-SEP-79 18:02:48 PAGE 001  
DK31VECTOR, TT0:DK31VECTOR

FORTTRAN IU U01C-03  
CORE-00K, UIC-E1,1233

1001 SUBROUTINE UNITU (R,S,U)  
1002 REAL R(3),S(3)

CC COMPUTE UNIT VECTOR AND MAGNITUDE  
CC

1003 U=SQRT(R(1)\*\*2 + R(2)\*\*2 + R(3)\*\*2)  
1004 IF(U.EQ.0.) RETURN  
1005 DO 40 I=1,3  
1006 DO 40 S(I)=R(I)/U

1007 C  
1008 RETURN  
1009 END

3-48

FORTTRAN IU STORAGE MAP

| NAME | OFFSET | ATTRIBUTES                 |
|------|--------|----------------------------|
| 1    | 000014 | REAL24 PARAMETER ARRAY (3) |
| 2    | 000016 | REAL24 PARAMETER ARRAY (3) |
| 3    | 000018 | REAL24 PARAMETER VARIABLE  |
| 4    | 000020 | REAL24 PARAMETER VARIABLE  |
| 5    | 000022 | INTEGER32 VARIABLE         |

FRI 28-SEP-79 18:02:44 PAGE 001  
DK31VECTOR, TT0:DK31VECTOR

FORTTRAN IU U01C-03  
CORE-00K, UIC-E1,1233

1001 SUBROUTINE DOTP (R,S,U)  
1002 REAL R(3),S(3)

CC COMPUTE DOT PRODUCT OF 2 VECTORS  
CC

1003 U=0.  
1004 DO 40 I=1,3  
1005 DO 40 U=U + R(I)\*S(I)

1006 C  
1007 RETURN  
1008 END

```

FORTRAN IV      STORAGE MAP
NAME  OFFSET  ATTRIBUTES
2      000014  REAL34      PARAMETER ARRAY (3)
3      000016  REAL34      PARAMETER ARRAY (3)
4      000018  REAL34      PARAMETER VARIABLE
5      000020  INTEGER32  VARIABLE

```

```

FORTRAN IV      U01C-03      FRI 28-SEP-78 18103187      PAGE 001
:ONE=0001, UIC=01,1233      D31VECTOR, T701=D31VECTOR

```

```

2001      SUBROUTINE CROSSP (R,S,T)
2002      REAL R(3),S(3),T(3)

```

```

      C      C      C      CROSS PRODUCT OF 2 VECTORS
      C      C      C
2003      T(1) = R(2)S(3) - R(3)S(2)
2004      T(2) = R(3)S(1) - R(1)S(3)
2005      T(3) = R(1)S(2) - R(2)S(1)
2006      C      RETURN
2007      END

```

```

FORTRAN IV      STORAGE MAP
NAME  OFFSET  ATTRIBUTES
2      000014  REAL34      PARAMETER ARRAY (3)
3      000016  REAL34      PARAMETER ARRAY (3)
4      000018  REAL34      PARAMETER VARIABLE

```

FORTRAN IV 0010-03 WED 24-OCT-79 12:08:19 PAGE 001  
 COPE-03N. JIC-E1.1233 ,T0:DK3:ROTATE  
 THIS ROUTINE DELIVERS THE SLANT RANGE AND CROSS RANGE VECTORS  
 CALCULATED FROM THE ANGLES READ FROM THE INPUT DEVICE NUMBER 5  
 THE ANGLES ARE INPUT IN THE ORDER YAW,PITCH,ROLL IN DEGREES.  
 THE CALCULATIONS ARE BASED ON THE EQUATION FOUND IN THE SRC DOCUMENT  
 SCR TP 77-284, DECEMBER 1977 RADAR IMAGE SIMULATION SYSTEM  
 F30602-77-C-0088 PDP 11/40 USER'S GUIDE  
 APPENDIX C, PAGE C-1

```

0001 SUBROUTINE ROTATE (RS,RC)
0002 REAL RS(3),RC(3)
0003 CONU = .0174533
0004 WRITE(5,10)
0005 READ(5,11)YAW,PITCH,ROLL
0006 FORMAT('INPUT YAW,PITCH,ROLL (REAL,SIGNED DEGREES)')
0007 10 FORMAT(F6.5,F6.5,F6.5)
0008 YAW=YAW*CONU
0009 PITCH=PITCH*CONU
0010 ROLL=ROLL*CONU
0011 RS(1)=COS(YAW)*COS(ROLL)-SIN(YAW)*SIN(PITCH)*SIN(ROLL)
0012 RS(2)=-SIN(YAW)*COS(PITCH)
0013 RS(3)=SIN(ROLL)*COS(YAW)+COS(ROLL)*SIN(YAW)*SIN(PITCH)
0014 RC(1)=COS(ROLL)*SIN(YAW)+COS(YAW)*SIN(ROLL)*SIN(PITCH)
0015 RC(2)=COS(YAW)*COS(PITCH)
0016 RC(3)=SIN(ROLL)*SIN(YAW)-COS(ROLL)*SIN(PITCH)*COS(YAW)
0017 RETURN
0018 END
  
```

FORTRAN IV STORAGE MAP  
 NAME OFFSET ATTRIBUTES  
 RS 000014 REAL\*4 PARAMETER ARRAY (3)  
 RC 000016 REAL\*4 PARAMETER ARRAY (3)

VARIABLE  
VARIABLE  
VARIABLE  
VARIABLE  
PROCEDURE  
PROCEDURE

REAL34  
REAL34  
REAL34  
REAL34  
REAL34  
REAL34

000116  
000122  
000126  
000132  
000000  
000000

COM  
PITCH  
ROLL  
COS  
SIN

RT-11 FORTRAN IV U018-08A MON 29-OCT-79 13:40:44 PAGE 001

```

0001 REAL D(256)
0002 INTEGER TTVI, TTVO, PARAFL(5), INFILE(5), PARAM(5), IMAGE(5)
0003 DIMENSION RS(3), RC(3)
0004 DEFINE FILE 30(S12,256,U,IAU)
0005 DEFINE FILE 31(1,256,U,IAU)
0006 WRITE(7,31)
0007 CALL ASSIGN(12, PARAFL, -1, 'RDO')
0008 WRITE(7,32)
0009 CALL ASSIGN(31, PARAM, -1)
0010 FORMAT(//, 'IMAGE FILE NAME', //)
0011 FORMAT(//, 'PARAMETER FILE NAME', //)
0012 FORMAT(//, 'ENTER DESTINATION FILE NAME', //)
0013 READ(12)NURDS,NRECS,RMAX,SRI,CRI,(RS(I),I=1,3),
X (RC(I),I=1,3)
0014 REWIND 12
0015 WRITE(7,50)NURDS,NRECS,RMAX,SRI,CRI,
X (RS(I),I=1,3), (RC(I),I=1,3)
0016 50 FORMAT(//, 'NURDS = ', I5, 'X', 'NRECS = ', I5, 'X', 'RMAX = ', F12.3, 'X',
X 'SRI = ', F8.3, 'X', 'CRI = ', F8.3, 'X', 'RS(1) = ', F8.3, 'X',
X 'RC(1) = ', F8.3, 'X', 'RS(1) = ', F8.3, 'X',
X (RC(I),I=1,3)
0017 WRITE(31,1)NURDS,NRECS,RMAX,SRI,CRI,(RS(I),I=1,3),
X (RC(I),I=1,3)
0018 WRITE(7,30)
0019 CALL ASSIGN(11, INFILE, -1, 'RDO')
0020 WRITE(7,32)
0021 CALL ASSIGN(30, IMAGE, -1)
0022 DO 45 I=1, NRECS
0023 READ(11) (D(K), K=1, NURDS)
0024 WRITE(30,19) (D(K), K=1, NURDS)
0025 CONTINUE
0026 REWIND 11
0027 STOP
0028 END

```

3-52

RT-11 FORTRAN IV STORAGE MAP

| NAME   | OFFSET | ATTRIBUTES          |
|--------|--------|---------------------|
| D      | 000006 | REAL14 ARRAY (256)  |
| PARAFL | 002006 | INTEGER12 ARRAY (5) |

|        |        |           |           |
|--------|--------|-----------|-----------|
| INFILE | 002020 | INTEGER12 | ARRAY (5) |
| NAME   | 002032 | INTEGER12 | ARRAY (5) |
| NAME   | 002044 | INTEGER12 | ARRAY (5) |
| PC     | 002056 | REAL14    | ARRAY (3) |
| PC     | 002072 | REAL14    | ARRAY (3) |
| TTYI   | 002436 | INTEGER12 | VARIABLE  |
| TTYO   | 002440 | INTEGER12 | VARIABLE  |
| JAU    | 002442 | INTEGER12 | VARIABLE  |
| JAU    | 002444 | INTEGER12 | VARIABLE  |
| ASSIGN | 000000 | REAL14    | PROCEDURE |
| MURDS  | 002446 | INTEGER12 | VARIABLE  |
| MURDS  | 002450 | INTEGER12 | VARIABLE  |
| PMAN   | 002452 | REAL14    | VARIABLE  |
| SRI    | 002456 | REAL14    | VARIABLE  |
| CRI    | 002462 | REAL14    | VARIABLE  |
| I      | 002466 | INTEGER12 | VARIABLE  |
| I9     | 002470 | INTEGER12 | VARIABLE  |
| K      | 002472 | INTEGER12 | VARIABLE  |
| X      |        |           |           |

## MISSION of Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control, Communications and Intelligence (C3I) activities. Technical and engineering support within areas of technical interest is provided to ESD Program Offices (POs) and other ESD elements. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.